

Open Research Online

The Open University's repository of research publications and other research outputs

Executive Attention, Action Selection and Attention-Based Learning in Neurally Controlled Autonomous Agents

Thesis

How to cite:

Garforth, Jason P. (2006). Executive Attention, Action Selection and Attention-Based Learning in Neurally Controlled Autonomous Agents. PhD thesis The Open University.

For guidance on citations see [FAQs](#).

© 2006 The Author

Version: Version of Record

Link(s) to article on publisher's website:
<http://dx.doi.org/doi:10.21954/ou.ro.0000573d>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

Executive Attention, Action Selection and Attention-based Learning in Neurally Controlled Autonomous Agents

A dissertation submitted in partial
completion of the requirements for the degree

Doctor of Philosophy in Computing

Jason P. Garforth



The Open University

FACULTY OF MATHEMATICS AND COMPUTING

Submitted - July 2007

Acknowledgements

I'd like to thank my wife Joanne who has had to put up with "The Robots" for the last 10 years and my daughter Lucy Elizabeth (who will be old enough to enjoy them in a few years)

I'd like to also thank the Sheffield Hallam University, The Open University and my supervisors Anthony Meehan and Chris Dobbyn as well as Sue McHale for helping to start and guide this work to completion

A big thank you to my friends & family who helped in various ways over the years, including my mum (Margaret), Keith & Lynn, Andrea & Nigel, Thomas, Daniel, Aimi, Emily, Dennis & Elsie, Jeremy L, Bev, Cath, Tim, Nick, Marie, Lynn, Justine, Peter A, Trevor S, Hillary, Martin W, Fiona & Pete M, Dave B, Dominic W, Stefan L, Ronnie M and Derek H & Angie

And IBM for not getting too much in the way

Related Publications

Garforth, J. P., McHale, S., and Meehan, A. (2006). Executive attention, task selection and attention-based learning in a neurally controlled simulated robot. *Neurocomputing*, (69):1923–1945

Garforth, J. P., McHale, S., and Meehan, A. (2004). Executive attention and action selection in a neurally controlled simulated robo. In Smith, L. S., Hussain, A., and Aleksander, I., editors, *Brain Inspired Cognitive Systems (BICS) 2004*

Garforth, J. P., McHale, S., and Meehan, A. (2003). Neural executive attentional control in robots. In *International Joint Conference on Artificial Intelligence (IJCAI-03)*.

Garforth, J. P., McHale, S., and Meehan, A. (2001b). Problems of attentional behavior in autonomous robotic systems. In Atkins, E. M. and Fesq, L. M., editors, *Robust Autonomy*, AAAI Spring Symposium, Stanford University, Ca., USA, pages 71–75. AAAI Technical Report SS-01-06, AAAI Press, Menlo Park, Ca., USA

Garforth, J. P., McHale, S., and Meehan, A. (2001a). Executive attentional control in autonomous robotic systems. In Zhong, N., Liu, J., Ohsuga, S., and Bradshaw, J., editors, *Intelligent Agent Technology Research and Development. Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology*, Maebashi, Japan 23-26 October 2001, pages 479–483. World Scientific Publishing, ISBN 981-02-4706-0

Garforth, J. P., McHale, S., and Meehan, A. (1999). Adjustably autonomous robotic agents. In Musliner, D. and Pell, P., editors, *Agents with Adjustable Autonomy*, AAAI Spring Symposium 1999, Stanford University, Ca., USA, AAAI Technical Report SS-99-06. AAAI Press, Menlo Park, Ca., USA, ISBN 1-57735-102-9

Executive Attention, Action Selection and Attention-based Learning in Neurally Controlled Autonomous Agents

Jason P. Garforth

email: jpg@janus.demon.co.uk

Abstract

I describe the design and implementation of an integrated neural architecture, modelled on human executive attention, which is used to control both automatic (reactive) and willed action selection in a simulated robot. The model, based upon Norman and Shallice's supervisory attention system, incorporates important features of human attentional control: selection of an intended task over a more salient automatic task; priming of future tasks that are anticipated; and appropriate levels of persistence of focus of attention. Recognising that attention-based learning, mediated by the limbic system, and the hippocampus in particular, plays an important role in adaptive learning, I extend the Norman and Shallice model, introducing an intrinsic, attention-based learning mechanism that enhances the automaticity of willed actions and reduces future need for attentional effort required for dealing with distractions. These enhanced features support a new level of attentional autonomy in the operation of the simulated robot. Some properties of the model are explored using lesion studies, leading to the identification of a correspondence between the behavioural pathologies of the simulated robot and those seen in human patients suffering dysfunction of executive attention

Contents

	<i>Page</i>
1. Introduction	1-1
1.1 The Problem of Behaviour	1-2
1.2 Framing the Research Question and Interdisciplinary Approach	1-5
1.3 Relevance of the Research	1-7
1.4 Organisation of the Thesis	1-8
	<i>Page</i>
2. Aspects of Method	2-1
2.1 Behaviour-based Artificial Intelligence	2-2
2.2 Structured-Case	2-4
	<i>Page</i>
3. Literature Review	3-1
3.1 Attention and Memory	3-2
3.2 Neuropsychological Models of Attentional Control	3-4
3.2.1 Baddeley and Hitch	3-5
3.2.2 Norman and Shallice	3-6
3.2.3 Duncan	3-9
3.2.4 Cowan	3-10
3.2.5 Baars	3-10
3.2.6 O'Regan and Noë	3-11
3.2.7 Selecting a Model for Practical Research	3-11
3.3 Attention-based Learning	3-12
3.3.1 Deriving an Attentional Mechanism for Learning	3-13
3.4 Related Research: Neurally Implemented Models of Attention	3-13
3.4.1 Cohen, Dunbar and McClelland	3-14
3.4.2 Gilbert and Shallice	3-14
3.4.3 Cooper	3-15

	<i>Page</i>
3.4.4 Taylor, Fragopanagos and Kasderidis	3-16
3.4.5 Franklin, Shanahan and Baars	3-18
3.4.6 Relationship of Previous Work to the Research Conducted	3-20
3.5 Summary	3-21
	<i>Page</i>
4. Model Design	4-1
4.1 An Overview of the Attentional Architecture	4-2
4.2 Major Functional Subsystems	4-4
4.2.1 The Reactive System	4-5
4.2.1.1 Reactive Sensors, Effectors and Perceptual Structures	4-5
4.2.1.2 Conflict Resolution and the Contention Scheduler	4-5
4.2.1.3 Episodic and Procedural Memory	4-6
4.3 Working Memory and the Supervisory Attentional System	4-9
4.3.1 Working Memory	4-9
4.3.2 The Supervisory Attentional System and Sub-functions	4-11
4.3.3 Working Memory and the SAS	4-12
4.4 The Affective System and Learning	4-13
4.5 High-level Summary Account of the Attentional Architecture	4-15
	<i>Page</i>
5. Detailed Design	5-1
5.1 Elements of Detailed Design	5-2
5.2 Recogniser Nodes and Pattern Matching	5-3
5.2.1 A Taxonomy of Recogniser Nodes	5-7
5.2.2 Bottom Up Recogniser Nodes	5-8
5.2.3 Top Down Recogniser Nodes	5-9
5.3 Working Memory	5-10
5.3.1 Memory Composition and Recogniser Types	5-11
5.4 The Structure of the Supervisory Attentional System	5-11
5.5 Summary	5-15

	<i>Page</i>
6. Implementation	6-1
6.1 Architectural Implementation	6-2
6.1.1 Compositional Elements of the Control Network	6-2
6.1.2 Clusters	6-3
6.1.3 Networks	6-6
6.1.4 Collections	6-6
6.1.5 Mappings	6-7
6.2 Developing Major Functional Elements of the Architecture	6-9
6.2.1 The Controlling Network	6-9
6.2.2 Episodic and Procedural Memory	6-9
6.2.3 The Contention Scheduler and Thalamus	6-12
6.2.4 Working Memory	6-14
6.2.5 The Supervisory Attentional System	6-17
6.2.6 The Affective System, Hippocampus and Learning	6-21
6.2.7 Visualising the Functional Components	6-22
6.3 Controller Training and Operation	6-23
6.3.1 Network Evaluation	6-23
6.3.2 Threading	6-24
6.4 Summary	6-25
	<i>Page</i>
7. Behavioural Properties	7-1
7.1 The Simulated Environment and Robot	7-2
7.1.1 Robot Sensors and Actuators	7-2
7.1.2 Observing the Controlling Network	7-4
7.2 An Instance of the Attentional Architecture	7-5
7.3 Normal Behaviour and Learning	7-6
7.3.1 Normal Behaviour in an Inexperienced Machine	7-7
7.3.2 Behaviour in an Experienced Machine	7-11

	<i>Page</i>
7.4 Lesion Studies in the Attentional Architecture	7-14
7.4.1 Lesioning the Attentional Architecture	7-14
7.4.2 Setting up the Simulation for Lesioning	7-16
7.4.3 Lesion Study A	7-17
7.4.3.1 Lesion Study A: Normal Level of SAS Modulation	7-17
7.4.3.2 Lesion Study A: High Level of SAS Modulation	7-18
7.4.4 Lesion Study B	7-19
7.4.5 Lesion Study C	7-21
7.4.6 Lesion Study D	7-23
7.4.7 Lesion Study E	7-24
7.5 Summary	7-25
	<i>Page</i>
8. Discussion	8-1
8.1 Normal Behaviour	8-2
8.2 Lesioned Behaviour	8-4
8.2.1 Lesion A	8-5
8.2.2 Lesion B	8-5
8.2.3 Lesion C	8-6
8.2.4 Lesion D	8-6
8.2.5 Lesion E	8-7
8.3 Summary	8-7
	<i>Page</i>
9. Concluding Discussion and Future Work	9-1
9.1 Relation of this Research to Open Questions in Cognitive Systems	9-2
9.2 Behavioural Evaluation and the SAS Generator	9-4
9.2.1 Further Work I	9-5
9.3 Attention, Autonomous Learning and Selective Advantage	9-6
9.3.1 Further Work II	9-8

	<i>Page</i>
9.4 Scalability and the Architecture of the SAS	9-8
9.4.1 Scalability	9-8
9.4.2 Architecture of the SAS	9-9
9.4.3 Further Work III	9-10
9.5 Robustness of Behaviour and Development of an Embodied Machine	9-10
9.5.1 Statistical Analysis of Observable Behaviour	9-10
9.5.2 Statistical Analysis of Control Traces	9-11
9.5.3 Developing an Embodied Machine	9-11
9.6 Summary	9-12
	<i>Page</i>
10. References	10-1
	<i>Page</i>
11. Appendix A	11-1
11 An Overview of the Sheep & Sheepdogs	11-1
11.1 Sheep & Sheepdogs	11-2
11.2 Autonomous Robots	11-2
11.2.1 Sheep 1, Sheep 2 and Sheepdog	11-2
	<i>Page</i>
12. Appendix B	12-1
12 XML for the Attentional Architecture	12-1
12.1 Attentional Architecture, Controller XML	12-2
12.2 Attentional Architecture, Controller Document Type Definition (DTD)	12-11
12.3 Creating SheepWorld3, the Attentional Architecture and this Thesis	12-13

Illustrations

2. Aspects of Method

	<i>Page</i>
<i>figure 2.1</i> The structured-case research method (Carroll and Swatman, 2000, p. 241) (Permission required)	2-5

3. Literature Review

	<i>Page</i>
<i>figure 3.1</i> The Baddeley & Hitch multi-component model of working memory	3-6
<i>figure 3.2</i> The original Norman & Shallice model [1986] adapted from [Norman & Shallice, 1986]	3-7
<i>figure 3.3</i> A simplified diagram of the Shallice & Burgess Supervisory attentional system decomposition adapted from Shallice & Burgess [1998]	3-8

4. Model Design

	<i>Page</i>
<i>figure 4.1</i> The original Norman & Shallice architecture for executive control of routine and non-routine behaviour	4-2
<i>figure 4.2</i> An overview of the proposed architecture for executive control of behaviour showing the reactive system consisting of episodic and procedural memory	4-3
<i>figure 4.3</i> The structure of the reactive links of episodic and procedural memory and their interconnections	4-8
<i>figure 4.4</i> An illustration of how episodic, working and procedural memories interact	4-10
<i>figure 4.5</i> Expanded diagram showing the speculated hierarchically structured episodic memory interacts with the presumed layered structure of working memory	4-11
<i>figure 4.6</i> Interactions between episodic, working memory and the Supervisory Attentional System	4-13
<i>figure 4.7</i> Learning in episodic memory	4-15
<i>figure 4.8</i> The Attentional Architecture showing the primary components and connection paths	4-16
<i>figure 4.9</i> The emotional and supervisory components of the Attentional Architecture	4-17

	<i>Page</i>
<i>figure 4.10</i> Interaction points between episodic, working and procedural memory	4-18
 5. Detailed Design	
	<i>Page</i>
<i>figure 5.1</i> A recogniser node, used as a basic building block for episodic, working and procedural memory subsystems within the Attentional Architecture	5-4
<i>figure 5.2</i> An illustration of how recogniser nodes interact with the contention scheduler and thalamic gates	5-5
<i>figure 5.3</i> A diagram of the proposed structure of episodic memory composed of recogniser nodes	5-6
<i>figure 5.4</i> The activity pattern of a ‘bottom up’ recogniser node, recognising an input pattern and becoming salient (active)	5-8
<i>figure 5.5</i> The activity pattern of a ‘top down’ recogniser node, being requested to become salient after recognising an input pattern	5-9
<i>figure 5.6</i> The speculated structure of recogniser nodes in working memory, interacting with other recogniser nodes in episodic and procedural memory	5-10
<i>figure 5.7</i> Recogniser types and their assumed primary location in each memory subsystem	5-11
<i>figure 5.8</i> An overview of a SAS Monitor and SAS Modulator	5-13
<i>figure 5.9</i> The activity pattern of an individual SAS Monitor and Modulator node	5-13
<i>figure 5.10</i> A modified version of <i>figure 5.7</i> , showing the memory areas potentially connected to and controlled by the Supervisory Attentional System	5-14
<i>figure 5.11</i> Interconnections between a recogniser node and a SAS Monitor/Modulator node shown as a modified <i>figure 5.2</i>	5-15
 6. Implementation	
	<i>Page</i>
<i>figure 6.1</i> An overview of the relationships between clusters (marked as C1 etc), the networks that contain them, collections that structure networks and mappings	6-3
<i>figure 6.2</i> A four input, four output Elman cluster, as shown in <i>listing 6.1</i> and displayed by SNNS	6-4

	<i>Page</i>
<i>figure 6.3</i>	The structure of the reactive links of episodic and procedural memory and their interconnections 6-10
<i>figure 6.4</i>	An overview of how the contention scheduler, thalamus gateways and memory subsystems interact 6-12
<i>figure 6.5</i>	The ‘thalamic gates’ used in bulk in the thalamus, protect shared information pathways from inappropriate access by competing recogniser nodes 6-14
<i>figure 6.6</i>	The simulation traces of the working memory node ‘collect_food’ taken from <i>SheepWorld3</i> , showing the willed intention of the machine as it generates a sequence of behaviours 6-16
<i>figure 6.7</i>	An overview of a SAS Monitor and SAS Modulator from <i>Chapter 5</i> 6-17
<i>figure 6.8</i>	The SAS monitor and modulator for the working memory observer node named ‘collect_food’ 6-18
<i>figure 6.9</i>	A simulation trace of the SAS cluster associated with the working memory node ‘collect_food’ generated by the <i>SheepWorld3</i> Navigator panel 6-20
<i>figure 6.10</i>	The process of suppressing an inappropriate node by the Supervisory Attentional System 6-21
<i>figure 6.11</i>	The same recognition cluster as illustrated in <i>figure 6.2</i> , shown at neuron level as part of procedural memory 6-22

7. Behavioural Properties

	<i>Page</i>
<i>figure 7.1</i>	The main canvas of SheepWorld3 (generated by the simulation), showing Penny the simulated robot together with a home location and a food object 7-2
<i>figure 7.2</i>	The robot Penny, depicting the traces (dashed lines) from the two sonar sensors 7-3
<i>figure 7.3</i>	Monitoring controller activity in context is done using the Navigator interface 7-4
<i>figure 7.4</i>	The instance of the Attentional Architecture used in the studies showing the nodes and their connections 7-5
<i>figure 7.5</i>	Initial configuration of the environment for the tactical test for distraction 7-8
<i>figure 7.6</i>	The final position upon completion of the experiment 7-8
<i>figure 7.7</i>	Traces from the simulation showing the activity of seven neurons from the model as the simulated machine moves from point B to point E 7-9

	<i>Page</i>
<i>figure 7.8</i>	In a second experiment, these traces illustrate the introduction of a more potent distraction to the machine whilst it is located at home 7-11
<i>figure 7.9</i>	The distraction is given maximum salience by placing it adjacent to the machine 7-12
<i>figure 7.10</i>	Traces from the third run of the simulation for a partially experienced machine 7-12
<i>figure 7.11</i>	A final run of the experiment using an ‘experienced’ controller taken from the end of the first experiment shown in <i>figure 7.5</i> 7-13
<i>figure 7.12</i>	Performance of a wholly inexperienced machine in different configurations of the test environment 7-13
<i>figure 7.13</i>	The five structural lesions (A - E) to the Attentional Architecture 7-15
<i>figure 7.14</i>	Lesions at the implementation level, illustrating the building blocks in episodic memory (adapted from <i>Chapter 6, figure 6.11</i>) 7-16
<i>figure 7.15</i>	The preliminary path taken by the inexperienced robot (Penny) and its associated neural traces prior to lesioning the controller 7-17
<i>figure 7.16</i>	The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study A 7-18
<i>figure 7.17</i>	A second run of lesion A , with the weighting between the SAS Modulator and the contention scheduler doubled 7-19
<i>figure 7.18</i>	The path taken by the inexperienced robot Penny during lesion study B 7-18
<i>figure 7.19</i>	The associated neural traces of <i>figure 7.17</i> during lesion study B 7-20
<i>figure 7.20</i>	The path taken by the inexperienced robot Penny during lesion study C 7-21
<i>figure 7.21</i>	The associated neural traces of <i>figure 7.19</i> during lesion study C 7-22
<i>figure 7.22</i>	The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study D 7-23
<i>figure 7.23</i>	The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study D when the lesion is introduced at the start of the experiment 7-24
<i>figure 7.24</i>	The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study E 7-25

11. Appendix A

	<i>Page</i>
<i>figure 11.1</i>	Sheep 1, Sheep 2 and Sheepdog 11-2

Tables & Listings

6. Implementation

	<i>Page</i>
<i>listing 6.1</i> A typical SNNS cluster file used in the Attentional Architecture	6-5
<i>listing 6.2</i> Building the network <code>gate_*</code> from the SNNS cluster file <code>thalamic_gateway.net</code>	6-6
<i>listing 6.3</i> Building a fragment of episodic memory using recogniser nodes	6-6
<i>listing 6.4</i> Defining the <code>thalamus_gateway.motor</code> collection	6-7
<i>listing 6.5</i> Using XML to define a layer within procedural memory	6-7
<i>listing 6.6</i> Mapping a control output from the contention scheduler to an enable input in the motor thalamic gateway	6-7
<i>listing 6.7</i> Creating information ‘pathway’ connections between episodic and procedural memory recogniser nodes	6-8
<i>listing 6.8</i> Using a “sensitize” element to allow neuromodulation of the mapping previously shown in <i>listing 6.6</i>	6-9
<i>listing 6.9</i> Using a “sensitize” element to allow neuromodulation of inhibitory connections between episodic recogniser nodes	6-9
<i>listing 6.10</i> The XML for implementing a fragment of the episodic and procedural memory structures shown in <i>figure 6.3</i>	6-11
<i>listing 6.11</i> The XML for implementing the structures illustrated in <i>figure 6.4</i>	6-13
<i>listing 6.12</i> Working memory including mappings as implemented using XML, based on <i>Chapter 5, figure 5.7</i>	6-15
<i>listing 6.13</i> The XML to create the SAS Monitor and Modulation cluster shown in <i>figure 6.8</i>	6-19
<i>listing 6.14</i> An XML listing showing how to control threading in the creation of the control architecture	6-25

9. Concluding Discussion and Future Work

	<i>Page</i>
<i>table 9.1</i> The (partial) contribution to the open questions posed in <i>Chapter 1</i> made within this thesis	9-4

Chapter 1

1. Introduction

This chapter establishes the context of the research programme described in this thesis. That context is a branch of behaviour-based artificial intelligence that brings together autonomous robotics, neuropsychology and neural computation in an attempt to explore the properties of behaviour-based systems. The chapter begins with a brief account of the history of autonomous, behaviour-based robotics which recounts what might be called the ‘strategic’ or ‘paradigmatic’ approaches and the problems they have eventually encountered. This leads to a statement of the central open question in the field that initiated this research, and of the initial insight that suggested a way in which to address this question. The chapter concludes with a summary of the contribution of the research presented in this thesis.

1.1. The Problem of Behaviour

In dynamic environments, where the unfamiliar or even the entirely novel may be encountered, ‘doing the right thing at the right time’ is important. Biology provides us with an existence proof, in the form of humans and other higher animals, that there is an effective, if sometimes fallible, ‘solution’ to this problem of action selection. In humans, successful action selection is believed to have two manifestations: automatic action selection and willed action selection [Norman and Shallice, 1986; Shallice, 1988]. (Automatic behaviour is also called non-voluntary, unconscious or routine behaviour and willed behaviour is called voluntary, conscious or deliberate behaviour.)

Automatic action selection ranges from wholly reflex actions (e.g., recoiling from something uncomfortably hot) through to actions that have become very well-learned (e.g., driving in familiar and unproblematic conditions). Automatic actions are the actions we perform naturally, without any apparent awareness. In contrast, willed behaviour involves deliberate, conscious, control of action (e.g., playing an unfamiliar piece of music or a new video game). Clearly, these terms define a spectrum within which behaviours can become progressively more automatic and the need for application of will becomes more intermittent.

Within Artificial Intelligence (AI), one of the main strategies for developing an understanding of behaviour-based systems has been to develop machines which exhibit more complex and robust behaviour. In the 1950’s, the technology of the early computing machines was used to connect simple circuits to sensors and actuators to create some of the first autonomous robots. Of particular note are Walter’s ‘turtles’ and ‘tortoises’ [Walter, 1953]. Walter turned to ‘Electro-biology’ to help him theorise about his research into the brain using the then new technology of electro-encephalography (EEG) and its ability to display the brain’s electrical response to stimuli such as light flashes, images and noises.

To expand his theories Walter created analogue versions of human neurons and embedded two of these together with a light sensor, a bumper-switch and two motor actuators (a drive wheel and a steering wheel) into a machine he called *Machina Speculatrix*. These machines were sensitive to light, sound and touch and they appeared to exhibit several emergent behaviours, including attraction to light or sound and returning to a recharging station (marked using light). Later machines could learn using a limited form of conditioning based upon a tone and a simple memory system.

Walter identified a number of aspects of animal behaviour that presented challenges and which remain pertinent over fifty years later, amongst which are:

- Typically, animals do not wait for things to happen to them, they are rarely passive
- Distractedness caused by sensory attractions in the environment seems to be ‘designed in’
- The ability to make a distinction between effective and ineffective behaviour
- Learning begins with recognising failure and is achieved through repetition

Although Walter knew that the brain was composed of functional units [Walter, 1953, p. 17], he also knew that the connections between neurons in these functional units and to other functional units [Walter, 1953, p. 77] were at least as important as the total number of neurons. The rise of the digital computer and its perceived correspondence with the human brain meant that Walter's research using analogue electronics for neurons and scanning technology such as EEG rapidly became unfashionable.

Braitenberg continued the development of comparable machines into the early 1980's [Braitenberg, 1984]. His machines exhibited behaviours through close, even direct, coupling of sensor to effector. Some such machines were able to integrate one or two basic behaviours, and to demonstrate basic learning. However, the problem of developing a control architecture that integrated a large number of low-level behaviours to achieve performance of high-level behaviour proved elusive and the focus of research effort had shifted to reflect developments in computing technology.

In the 1970's and 1980's research focussed on deliberative machines [Newell, 1990; Nourbakhsh et al, 1995] which received information from the environment, reasoned rationally about that environment, often using an internal model based on rules or axioms, to arrive at a plan which was then executed. Associated monitoring and correction routines sought to ensure its successful completion. Formally, these reasoning engines represented the world, objects, and sometimes even themselves, as internal symbols and they could manipulate and reason about these symbols to produce seemingly intelligent behaviour. This method of producing intelligence through manipulating symbols was the basis of the Physical Symbol System Hypothesis [Newell and Simon, 1976], which stated that formal symbol manipulation is the only mechanism that can produce general intelligent behaviour in machines. Symbolic manipulation was later characterised by Beer [Beer et al., 1990] as Classical AI. An example of a classical AI program that is still being actively developed and researched is SOAR [<http://sitemaker.umich.edu/soar>]. This system was originally written in the late 1980's by Laird, Rosenbloom & Newell [Newell, 1990] as a tool to explore general cognition.

While such systems were relatively good at reasoning about highly specific domains, they had several drawbacks when deployed in autonomous machines, most notably poor real-time performance in a dynamic environment and poor performance in the face of the unfamiliar. By the mid 1980's it was being suggested that there were fundamental limitations on contemporary computing technology in respect of developing intelligent and autonomous machines. In particular, the so-called Frame Problem [McCarthy, 1963; McCarthy and Hayes, 1969] posed the seemingly insurmountable consequences of continually increasing the number and specificity of axioms maintained by a machine. Dennett suggested that, in order to avoid the frame problem (as distinct from solving it), a system would need “a way of genuinely ignoring most of what it knows and operating with a well-chosen portion of its knowledge at any one moment” [Dennett, 1998, p. 197].

From the mid 1980s to early 1990s two developments in computing technology led to a renewed engagement with the problem of integrating low-level behaviours to exhibit robust, higher-level

behaviour in machines. Firstly, Brooks introduced his subsumption control architecture, which appeared to offer a design strategy for integrating successive levels (layers) of behaviour to realise robust performance in dynamic environments [Brooks, 1986]. And secondly, the connectionist manifesto of Parallel Distributed Processing (PDP) [Rumelhart and McClelland, 1986] which exploited the (re)discovery of efficient algorithms for training multi-layer neural networks [Rumelhart, Hinton and Williams, 1986].

Brook's subsumption architecture for autonomous vehicle control used behaviour layers or levels of competence. Successively higher levels of competence contain (subsume) lower levels of competence. Layers could be built as completely separate components and simply added to existing layers to achieve an overall level of competence. However, the experience of building real robots with the subsumption architecture identified issues with the architecture which arose when trying to manage large collections of finite state machines and synchronisation [Brooks, 1988]. Some of these problems were corrected with the development of the Behaviour Language [Brooks, 1990a], a language for writing large collections of subsumption architecture finite state machines. However, although several researchers built complex reactive machines using subsumption, adding behaviours, deliberative layers and learning proved to be more intractable [Brooks, 1990b; Ferrell, 1993; Cliff and Ghanea-Hercock, 2006].

Neurally controlled robots demonstrated success in respect of achieving low-level autonomous functions such as obstacle avoidance, wall-following, and map building but, in common with the subsumption approach, there was lack of progress in developing architectures for high-level functions and control. One response was to model ever more basic behaviour, often guided by insect neuroanatomy and neurophysiology, [Beer, 1990]. A second response was to draw upon animal ethology to design the control systems based upon large collections of parallel processes (often but not always neural networks) which connect to sensors, motors and other processes [Cliff et al., 1993; Kein, 1992; Lambrinos and Scheier, 1995]. The first of these strategies had moved away, temporarily, from the goal of higher-level function. The second achieved a measure of further success; enabling machines to combine low-level behaviours to form limited high-level behaviours [Mataric, 1996].

By the mid 1990s it was recognised that autonomous machines developed within one or other of these new approaches seemed to share common behavioural problems in respect of all but the most-simple tasks. In a review of the field Maes [1994] identified a number of common behaviour-related problems exhibited by many autonomous robots (and which bear a striking resemblance to the problems identified by Walter). These were:

- Excessively frequent and inappropriate changes of behaviour, appearing as distractedness or as indecision
- Inappropriate persistence of one behaviour when another is seemingly more appropriate or desirable

- Repetitive/cyclical behaviour in which the robot appears to lack any awareness of the need to revise its current strategy in order to make progress

For many developers of autonomous machines, the response to the lack of progress in developing effective high-level controllers using biologically inspired techniques was to turn to hybrid architectures which featured neural and subsumption control systems for low-level, automatic actions such as obstacle avoidance and wall and map following, together with classical planning and scheduling systems for strategic control of willed actions. Hybrid control systems offered a pragmatic strategy for enhancing performance, for example, by switching between high-level and low-level behaviour, or maintaining a high degree of separation between their functioning.

However, hybridity leaves unexplored the outstanding question of how integration of low-level behaviours to achieve high-level behaviour is achieved using neural architectures such as those in animals and humans. The inaugural edition of the journal *Artificial Life*, formulated two of the main open problems to be solved [Maes, 1994, p. 138]:

- The problem of action selection: given a set of time varying goals, a set of actions that an agent is capable of performing, and a set of sensor signals, which action(s) should an agent execute at each instant so as to make progress towards one or more of its goals?
- The problem of learning from experience: how can an agent modify its behaviour producing structures and processes so that it improves performance and acquires new competencies?

The action selection problem is governed by the practical, real world and real time constraints that apply to a goal-directed embodied agent. Such an agent has specific sensors, specialised effectors, and limited computational resources, which together serve to constrain the scope of perception and action. Related to the problems posed by Maes, Brooks [2002, p/ 181] makes clear in a discussion about the adaptive limitations of current robots and artificial simulations, that these systems are still a long way from emulating biological systems.

1.2. Framing the Research Question and Interdisciplinary Approach

In looking to address the questions above, the point of departure for the research presented here was a recognition that the behavioural problems of many autonomous robots (as described above) appeared in considerable measure to correspond to behavioural pathologies of humans who suffer lesions to the frontal areas of the brain associated with control of attention. In humans, difficulty in managing non-routine action is frequently associated with dysfunction of the pre-frontal cortex, in an area which is functionally associated with the Supervisory System [Baddeley, 1993], the Supervisory Attention System [Shallice, 1988] or the Executive [Parkin, 1996]. Examples of these behaviours include:

- Distracted behaviour or ‘capture errors’; an inability to suppress a strongly triggered, but inappropriate behaviour [Shallice, 1988]
- The inability to act (akinesia) - attributed to an inability to resolve selection between competing behaviours [Robbins, 1991]
- Persistence of an inappropriate behaviour (stereotypy or perseveration), a failure to notice significant cues that should result in the expression of a different behaviour [Shallice, 1988]

It was also recognised that, in many respects, human attention conforms closely to Dennett's requirement for “a system that reliably ignores what it ought to ignore in a complex environment with a large range of available actions and outcomes” [Dennett, 1998, p. 197].

Thus, the central hypothesis within this thesis is that the problem of action selection may be approached by an elaboration and implementation of a neural control system incorporating supervisory attention. In undertaking research on how this might be achieved, a possible mechanism by which progressive and autonomous learning of automatic behaviour emerged and gave rise to a secondary hypothesis: that attentional effort is the basis of an intrinsic reinforcement signal that induces automatic learning of tasks which initially require attentional resources.

In considering action selection, it is possible to distinguish two sub-problems: action specification and action expression. Action specification concerns the formulation of the actions to be taken in response to the goals of the agent, its internal state and the perceived state of the world. In humans, and probably other higher animals, this process of action formulation is not a single or uniform process; it involves parallel sub-processes, each with distinct but interfacing neural pathways. The processes involved in action specification are capable of rendering many actions relevant at one and the same time. Where specified actions make demands upon wholly distinct effectors they can be expressed synchronously. However, many actions are likely to compete for expression via the same effectors. Actions which make equivalent demands upon the same effectors (albeit to different ends) may also be expressed synchronously. Where actions make contradictory demands upon the same effectors one must be selected over another. Action expression concerns the resolution of the demands of all specified actions so that coherent behaviour is expressed by an agent via its effectors.

Upon commencement of the research programme presented here, the problem of action expression (as opposed to action specification) was being addressed with considerable success by others; the focus being on modelling specific neural structures [Cooper and Shallice, 1997; 2000], and especially the basal ganglia [Houk et al., 1995; Presott et al, 1999; 2006; Gurney, 2001a; 2001b]. However, there was no neural controller that integrated executive attention into an architecture for both specification and expression in an autonomous machine, real or simulated.

The main obstacle to a strategy which seeks to draw upon neuropsychology and neuroanatomy to inform the development of an integrated control architecture for action specification using executive attention is the fact that these disciplines are themselves far from any degree of certainty about how such systems operate. Even when there is relative consensus at the functional level of description, there is often much less agreement at the anatomical level of description. In particular, this reflects a long-standing debate about the extent to which attention and the control of attention is localised or distributed. The current view, largely informed by imaging studies, synthesises these two perspectives, concluding that the major descriptive categories of brain function (e.g. vision or memory) are realised by interconnected networks of specialised functional centres, but that the functionality of the majority of such centres is poorly understood [Edelman, 2006].

1.3. Relevance of the Research

Upon commencement of the writing of this thesis (2006), the UK Government's Office of Science and Innovation (OSI) published their Foresight Project entitled 'Cognitive Systems'. This project sought to evaluate the scope for "reconnecting computing and AI with life sciences and neuroscience as a strategy for understanding complex biological systems so that their properties and mechanisms could inform computing and engineering" [Morris et al., 2006, p. 46]. The project reiterated the continued relevance of research on action selection. In relation to action specification, two related open problems were identified [Morris et al., 2006, p. 46]:

- Developing architectures that allow modules (actions or behaviours) to dynamically and automatically reconfigure themselves
- Enabling agents to generate novel behaviours and, through learning, integrate these new behaviours with its functioning set of behaviours and its strategic objectives.

In considering how these problems might be addressed within cognitive and computational neuroscience two more specific open questions are posed [Barnard and Redgrave, 2006, p. 129]:

- How is action selection influenced or determined by attentional mechanisms working upon information about current external states of the environment and internal states of the agent?
- How does information in memory interact with those current states?

This thesis can be seen as making a (partial) contribution to these questions. The principal contribution of the thesis is the formulation and qualitative validation of a neural control architecture for an autonomous machine (a simulated robot) in which executive attention contributes to otherwise automatic action specification (and action expression) and to learning. As noted, the details of equivalent architectures in human and animal systems are the subject of ongoing debate, not only as to how they map onto anatomical structures, but even as to whether the functional distinctions are reflected in anatomical distinctions. As Morris et al. [2006, p. 198] observe, recent research has begun

to call into question the sharp distinctions traditionally made between perception, memory and action, and between attention systems and (some forms of) memory. Thus, any specific control architecture for a machine such as the one presented here, represents a hypothesis. The space for hypothesising is bounded by two constraints: the first is the need to encompass the scale of function and integration required; the second by the need to remain in accord within current boundaries of knowledge of neuropsychology and neuroanatomy. Given current knowledge, there is inevitably a trade off between these two needs. The work which underlies this thesis relates to macro-function rather than micro-function. It brings together major functional sub-systems for action specification (which includes executive attention), memory, selective action expression and, latterly, learning.

In order to reinforce the validity of the hypothesised control structure, it focuses upon the extent to which ‘macro’ properties align with those of the biological systems (animals and humans) which inform it. Clearly, it is important to demonstrate that the system performs ‘as expected’ at some high level of description. Drawing upon the established tradition of lesion studies within neuropsychology and neuroscience, it also examines more subtle correspondences between the modelled attentional control system and biological systems, where ‘correspondence’ entails the notion that, in addition to normal function, the pathologies should also be comparable.

1.4. Organisation of the Thesis

The remaining sections of this thesis are organised as follows;

- *Chapter 2* considers issues of methodology. It does so by locating the research within a particular ‘philosophical’ perspective on behaviour-based AI. It also gives an account of the progressive iterative development of the concepts and models that have produced the current stage of development of the work.
- *Chapter 3* reviews the major neuropsychological models of executive attention, and of attention-based learning, before examining related experimental work. The chapter justifies the adoption of a particular model of attention as the starting point for the research and it distinguishes the work done here from related work by others.
- *Chapter 4* develops a high-level description of the system to be implemented and explored.
- *Chapter 5* progresses to a lower-level of design to produce elements from which a working system can be developed.
- *Chapter 6* details the strategy for implementation.
- *Chapter 7* presents details of its evaluation, and, in particular, the results of a series of lesion studies designed to explore its properties.
- *Chapter 8* provides a discussion of the results, relating them directly to behavioural pathologies of humans and animals.
- *Chapter 9* charts some outstanding issues and points to future work.

Chapter 2

2. Aspects of Method

This chapter gives an account of two complementary aspects of method that have underpinned the research presented in this thesis. The first aspect describes the underpinning epistemological perspective, that of behaviour-based artificial intelligence. This establishes the boundaries that distinguish between concepts and methods that are consistent with the chosen model of enquiry and those that are inconsistent. The second aspect of method concerns the structure of the research activity that has produced the contribution to knowledge presented here: *Structured-Case*. These complementary aspects of method have been combined in a way which has sought to ensure that both the nature of the work done and the directions in which the work evolved were always rooted in existing knowledge and theory and in sound engineering practice in order to develop further knowledge, understanding and theory.

2.1. Behaviour-based Artificial Intelligence.

The research programme that led to the work presented in this thesis was initiated in the mid-to-late 1990s, a period shortly after significant developments had taken place in the study of artificial intelligence. Behaviour-based artificial intelligence and neuroethology were emerging as new sub-disciplines of Artificial Intelligence (AI) and an important step was the establishment of principles and methods for conducting research within them. Behaviour-based artificial intelligence is concerned with understanding behaviour exhibited by artificial systems, reflecting the way psychology is concerned with understanding behaviour animals and humans. An early and important statement of the epistemological foundations of behaviour-based AI appeared in the inaugural issue of the journal *Artificial Life* [Steels, 1994] and this has served to guide much of the work described here.

Behaviour is defined as an observable regularity in the interaction dynamics of an agent's processes with those of the environment in which it is situated [Steels, 1994, p. 76] and the objective of behaviour-based AI was identified as developing a better understanding how high-level behavioural function arises through the interplay of behavioural components which individually exhibit low-level functionality. The level of description is 'behaviour' and 'behavioural components' from the perspective of an external observer (as in animal ethology or behavioural psychology). The level of explanation is a theory that accounts for observable behaviour.

In analysing and describing behaviour systems, a distinction is made between functionality, behaviour, mechanism and component:

- Functionality can be considered synonymous with purpose, task, goal and competence
- Behaviour is the observed regularity arising from the interaction between an agent and its environment. One or more behaviours (expressed in parallel and/or temporal sequence) contribute to functionality
- Mechanism is a technique for establishing a behaviour, such as direct coupling of sensor and actuator (e.g. photo-taxis as exhibited by a *Machina Speculatrix* [Walter, 1953] or a Braitenberg [1984] machine), or supervised learning in a neural network for wall-following
- Components are processes or physical entities which realise a mechanism and include sensors, actuators, programs and data structures

Methodologically, behaviour-based AI entails the construction and observation of artificial systems. The models it constructs may distinguished from those of traditional knowledge-based AI by virtue of a strong biological orientation and particularly by a significant use of neural networks to implement important components of systems [Steels, 1994, pp. 78-79].

The units of investigation are behaviour systems. Behaviour systems structure behaviour mechanisms, the latter being a principle or technique for establishing a particular behaviour using components.

Components include sensors, actuators, data structures, programs, communication hardware and software. An example of a mechanism is a specific coupling (hard or soft) between specific sensors and specific actuators.

In developing and investigating behaviour systems four organisational themes which relate behaviour components are common: cooperation, competition, architecture, and reinforcement (learning/adaptation). Each of these themes will be seen to feature prominent in the research described in subsequent chapters:

- Complementary behavioural components cooperate (combine) to realise higher-level behaviours
- Contradictory behavioural components compete for expression at the same effectors
- Observable behaviour, both reactive and attentionally controlled, is realised through the architectural relations between functional subsystems
- The architecture includes an intrinsic reinforcement mechanism which drives adaptation

Steels identified three means by which research scientists in artificial intelligence could build theories about behaviours or intelligence:

1. Mathematical models that relate observed variables in a system or systems to hypothesised variables. These can then be used in various ways, for example to calculate expected outputs from a given set of inputs and measure these against what the actual system produced
2. Computational models (or simulations) that consist of an algorithm working over a set of data structures. Executing the algorithm manipulates the internal data structures and if the corresponding output correlates with observed natural behaviour the model can be used as a theory of how the process may operate
3. Artificial models comprise a physical device which is built to replicate a set of observable phenomena. The way, in which an artificial model is designed, its components are built, and the way it interacts with the world around it constitutes the theory

The 'outputs' of a model of executive attention-based behaviour, as developed in this thesis, are expressed behaviours. Accordingly, it is appropriate to choose a modelling approach that allows for observation of these behaviours in a relatively naturalistic fashion. Both simulation and artificial modelling offer routes to building models of executive attention which allow relatively naturalistic observation of behaviour. Work undertaken prior to that presented in this thesis resulted in series of computer simulations and physical robots. However, in choosing to develop a neural model of executive attention which required a large-scale modular neural network it was necessary, from a pragmatic point of view, to use digital rather than analogue technology. This choice serves to blur the distinction between simulations and physical devices (in the senses deployed by Steels) because in

both a simulated robot and a physical robot the controller is realised as software running on a digital computer. Thus the notion of simulation is probably the more accurate way in which to characterise the research that has been conducted.

The behavioural system presented here consists of a software system that constitutes a neural model of executive attention and automatic (reactive) action selection controlling a simulated robot. The functionality of the system is the task of foraging, which requires the performance of an appropriate sequence of behaviours and repetition of: locate food, collect food, return to home, deposit food. The system also incorporates an intrinsic attention-based learning signal so that performance in this task improves with experience as the system learns to overcome a significant task-related distraction.

The main purpose of the research is to analyse and understand the properties of a system which relates executive attention and reactive (automatic) behaviour. As Steels puts it, “the model is a theory of how the process that is the behaviour system may operate”. In order to explore that process in operation, the failure modes of the model are explored. (In fact, as with even simple neural systems, ‘failure’ is better understood as the way in which the performance of the model ‘gracefully degrades’ as a result of damage to its components.)

The method used follows an approach deployed by Hinton, Plaut and Shallice [1993] when investigating forms of dyslexia in a neural model of semantic memory. In this method ‘lesion studies’ are conducted within which specific elements of the model are damaged and the resulting behavioural effects are observed. Based on their work and other accounts [Allport 1985; Rumelhart and McClelland, 1986; Quinlan 1991] Plaut and Shallice [1993] suggest that modular connectionist networks may offer a mechanistic account of areas in psychology and as a method for developing models of cognitive processes. Lesioning these models can also be done in a gradual, formal manner without having to make assumptions about the damage caused by the lesion and can result in a set of ‘in-between’ states which can be more informative about the behaviours being investigated.

The behavioural effects of lesions may be considered in light of behavioural pathologies seen in humans suffering lesions to areas of the brain associated with executive function. In this way, the model, as a theory, can be explored, confirmed, refuted, or refined.

2.2. Structured-Case

In contexts where problems and phenomena are ill-defined, and where the goal of exploration is to build or contribute to theory, it is helpful, not to say important, to structure activity in a way that enables developing ideas and understanding to remain grounded (for example, within the terms of behaviour-based AI as considered above).

An approach to structured enquiry that achieves the above is Structured-Case [Carroll and Swatman, 2000]. The term ‘case’ is used to specify the object of study; for instance, a person, a group, a project,

an organisation, a process. In this research, the object of study is the architecture for executive attentional control of behaviour and attention-based learning. Originally developed in the field of Information Systems, the goal of structured-case is to produce new, revised or refined knowledge and theory that describes relationships between meaningful and important concepts but is demonstrably rooted in observation. Structured-case features (as represented diagrammatically in *figure 2.1*) :

- An evolving conceptual framework representing the current state of a researcher's aims, theoretical foundations and understandings. The researcher begins with an initial conceptual framework based upon prior knowledge and experience (CF₁) iteratively revising it (CF_i) until their enquiry terminates (producing CF_n)
- An iterative research cycle featuring: data collection, analysis, (re)interpretation and synthesis
- An ongoing literature-based scrutiny which is used to compare and contrast the evolving outcomes of the enquiry with extant literature which may either support or contest them

figure 2.1 The structured-case research method (Carroll and Swatman, 2000, p. 241) (Image removed by editor as copyright not cleared)

Structured-case recognises that in many projects, including research projects, one starts with a basic, undeveloped and possibly incorrect understanding of a problem together with a methodological framework in which to develop that understanding a little more. Through observation, enquiry and/or experiment, followed by reflective analysis, one is lead to an improved, deeper and/or more comprehensive understanding. One will then be in a better position to take the next step forward. This approach to structuring activity:

- Recognises that one comes to a problem with some initial preconceptions, which may or may not turn out to be well-founded

- Recognises that engagement with the literature is an ongoing process that continues to guide the research
- Acknowledges that one can't always predetermine the attainment of the eventual outcome or goal and accepts that a successful project is one which makes a discernible and helpful contribution to other peoples' understanding of a problem area

Structured-case doesn't aim to be a methodology, as it doesn't prescribe what methods to use at each step; methods will need to be chosen to suit the characteristics of the research. In terms of the research presented here, the initial conceptual framework can be considered as:

- The observation (intuition) that the behavioural limitations of reactive, neurally controlled robots mirrored the behavioural pathology of human patients with deficits of executive attention
- The production rule system architecture for a Supervisory Attention System described by Norman & Shallice, 1986; Shallice, 1988]

The development of the conceptual framework can be traced through the associated publications (see page ii):

- Garforth, J. P., McHale, S., and Meehan, A. (2006). Executive attention, task selection and attention-based learning in a neurally controlled simulated robot. *Neurocomputing*, (69): 1923–1945
- Garforth, J. P., McHale, S., and Meehan, A. (2004). Executive attention and action selection in a neurally controlled simulated robot. In Smith, L. S., Hussain, A., and Aleksander, I., editors, *Brain Inspired Cognitive Systems (BICS) 2004*
- Garforth, J. P., McHale, S., and Meehan, A. (2003). Neural executive attentional control in robots. In *International Joint Conference on Artificial Intelligence (IJCAI-03)*.
- Garforth, J. P., McHale, S., and Meehan, A. (2001b). Problems of attentional behavior in autonomous robotic systems. In Atkins, E. M. and Fesq, L. M., editors, *Robust Autonomy*, AAAI Spring Symposium, Stanford University, Ca., USA, pages 71–75. AAAI Technical Report SS-01-06, AAAI Press, Menlo Park, Ca., USA
- Garforth, J. P., McHale, S., and Meehan, A. (2001a). Executive attentional control in autonomous robotic systems. In Zhong, N., Liu, J., Ohsuga, S., and Bradshaw, J., editors, *Intelligent Agent Technology Research and Development. Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology*, Maebashi, Japan 23-26 October 2001, pages 479–483. World Scientific Publishing, ISBN 981-02-4706-0
- Garforth, J. P., McHale, S., and Meehan, A. (1999). Adjustably autonomous robotic agents. In Musliner, D. and Pell, P., editors, *Agents with Adjustable Autonomy*, AAAI Spring Symposium 1999, Stanford University, Ca., USA, AAAI Technical Report SS-99-06. AAAI Press, Menlo Park, Ca., USA, ISBN 1-57735-102-9

It also included the development of two generations of simulation systems and physical machines (see *Appendix A*).

The literature that has guided the work is largely as described in *Chapter 3*. The evolving conceptual framework and its implementation have been closely informed by ongoing reference to literature on macro neuroanatomical structure of attention and memory systems and has been guided particularly by the emerging understanding of attentional dysfunction in the neuropsychological literature.

As a consequence of this approach, much of the literature cited throughout the thesis has appeared contemporary with, and often after, the research it is related to had been concluded. Certainly this is the case for literature published since 2003, when the emphasis of the research programme shifted from analysis and design of the architecture to its final implementation and evaluation, and subsequent publication [Garforth et al., 2004; 2006].

Chapter 3

3. Literature Review

This chapter presents the literature that directly relates to neurologically-based models of executive attention and thus lays the basis for the development of the research programme that is the subject of this thesis. The chapter is organised into four main sections:

- The first section presents an account of the main features of human executive attention and of memory. This account is not itself a model, rather it presents phenomena which are widely perceived as being characteristic of attention and memory. It is introduced here as a basis upon which to evaluate a range of models of executive attention, theoretical and empirical, that are covered in the subsequent parts of this chapter
- The second section provides an account of the leading theoretical neuropsychological models which have shaped the study of attention for the past half century and includes a statement of what might be considered the current consensus with respect to distinctive features of executive attention as given in a recent review of the field. This section concludes with an assessment of which model offers a foundation for the practical work of the research programme
- The third section examines the relationship between attention, adaptation and learning
- The final section considers related work by other researchers developing applied models of executive attention using a neural or quasi-neural approach. Much of the work is rooted in one of the theoretical models described in the earlier section, but others adopt a more pragmatic approach

As indicated in *Chapter 2*, there has been a continuous engagement with relevant literature throughout the programme of work. This literature has been used to test emerging perspectives, inform design decisions and to evaluate experimental results. Not all of this literature is considered here. Rather it is considered within the thesis where it relates directly to the context in which it was referenced. For example, the detailed design of many of the neural subsystems and the connections between them, make reference to specific sources that have shaped their design.

3.1. Attention and Memory

In a review of what might be considered the current consensus on features of human executive attention, LaBerge [1999] identifies three characteristics that should be evident in any model: selection, priming and use of memory for sustained task focus. In the context of action selection and behaviour these three properties can be described as follows:

- Selection concerns the expression of a willed action over a more salient, automatically selected, action. Here, the notion of salience is most closely connected to environmentally derived stimuli in the degree to which they accord with the relevance of contending actions. However, it may also derive from internal or innate drives. For example, the salience of feeding behaviour is determined both by the availability of food in the environment and by a sense of hunger or satiation. Willed action selection involves the application of an internally derived attentional signal which potentiates the desired behaviour and attenuates the automatic behaviour. This results in an increased likelihood of expression of the less salient act in preference to the more salient act. The attentional effort needed to will one familiar action in place of another is usually intermittent, or even momentary. The willing of wholly unfamiliar actions may require more persistent attention
- Priming concerns anticipation of future perception and action. It too is associated with an internally derived attentional signal. On this occasion, the potentiation is less likely to result in the immediate expression of the behaviour, rather it enhances the salience of the behaviour so that, when the anticipated circumstances arise, there is a greater likelihood that the anticipated task will be selected. Priming is associated with enhanced speed of switching to an anticipated task
- Use of memory for sustained task focus. Memory maintains focal, task-related information which includes selected sensory information and goals/intentions. Memory is particularly important when resumption of a suspended task requires recall of some past state that can no longer be inferred from observing the current state of the environment

The above account of the use of memory to sustain task focus points to the importance of the relationship between memory and the control of attention. Accounts of how memory is structured and distributed in the brain differ considerably, but functional accounts of memory exhibit a greater degree of consensus [Morris, Tarassenko and Kenward, 2006].

A common categorisation of memory distinguishes between short-term working memory and long-term memory. Working memory holds limited amounts of information for limited amounts of time. However, human memory capacity clearly exceeds this particular form of memory. Long-term memory endures, seemingly without conscious effort. Long-term memory is routinely subdivided into episodic (also biographical) memory, familiarity-based recognition memory, semantic memory, procedural (or skill) memory, and emotional (also value or affective) memory, not only because these

distinctions serve descriptive purposes in relation to function, but because there is evidence of anatomical separation of these forms. In the context of attentional control of behaviour as considered here, the most important to consider here are episodic, procedural and working memory:

- Episodic memory stores memories of biographical events. It does not appear to store event information that has not been attended to and is not permanent. Anatomically, episodic memory appears to be distributed across a number of structures. Memory encoding and recall are associated with the prefrontal lobe and part of the parietal lobe, but other sub-functions are anatomically distributed
- Procedural (or skill) memory encodes information on the performance of motor actions. The highest levels of skill are acquired through repeated practice. Actions which initially require deliberate and enduring attention become progressively well learnt to the point where they may be performed automatically. Procedural memory is persistent; well-learned skills such as swimming or bicycle riding are seldom forgotten. Anatomically, procedural memory is associated with the pre-motor and motor cortex. The ability to perform skills smoothly is associated with orderly action expression mediated by the basal ganglia
- Working memory holds limited amounts of information for limited amounts of time. People are able to maintain some deliberate focus on about seven tasks or issues at any one time [Miller, 1956]. Working memory is believed to feature a number of memory components, including willed goals and intentions, salient biographical episodes from episodic memory, unfamiliar actions from procedural memory that are wilfully combined and expressed

These memory systems are architecturally related and interdependent [Fuster, 2003, p. 235]. Episodic and procedural memories and their neural representation seems hierarchically organised. Both structures 'blend' into cortical structures in pre-frontal cortex, the presumed site of working memory, giving rise to the notion that the focus of attention is the focus of representation [Fuster, 2003, p. 237]. Active memory is maintained by the circulation of activity through sub-cortical (basal ganglia and thalamus) and limbic systems as well as recurrent and reciprocal connections in memory itself (perceptual and motor), hence relevant motor actions will tend to maintain active perceptual memory.

The pre-frontal cortex (working memory) appears to have the ability to organise and structure temporal behaviour, based on information and events that are separate in time. This ability seems supported by linked but distinct systems that are co-located in pre-frontal cortex:

- A representation of past perceptions and events that is closely associated with perceptual attention
- A representation of anticipated actions and events (the preparatory set) which primes perceptual and motor systems and appears closely associated with high and low level planning

Beyond the features of attention and its related memory systems considered above, it is appropriate also to address the relationship between attention and learning. In humans (and some animals), tasks which are initially novel and demanding of executive attention, if encountered and attended to frequently, or addressed with sufficient sustained attentive effort, become learnt to the point where they become automatic, needing an expression of will on rare occasions [Baars, 1993]. This suggests an additional characteristic of attention-based behaviour which can be added to those listed by LaBerge, above:

- Attentional effort leads to increased automaticity in task performance. A task which initially needs sustained attention comes to need intermittent, and then momentary or transient effort, until it is ‘automatic’ (e.g., learning a piece of music, through practice, to performance standard)

The above account of the features of attention and its relation to forms of memory provide a basis upon which to consider models of attentional control.

3.2. Neuropsychological Models of Attentional Control

“Every one knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalization, concentration, consciousness are of its essence. It implies withdrawal from some things in order to deal effectively with others, and is a condition which has a real opposite in the confused, dazed, scatterbrained state which in French is called *distracted*” [James, 1890]

When William James (sometimes described as the ‘father’ of psychology) says that everyone knows what attention is, he might be considered both right and wrong. He is right that every (healthy) person knows it through experience, but more than 100 years after his observation its underlying mechanisms remain very poorly understood. James’s statement contains many of the key elements of the current appreciation of the phenomenon. ‘Taking possession by the mind’ conveys the sense of becoming aware of something that had previously been unnoticed. ‘Focalisation’ conveys selection of important detail and ‘withdrawal’ that of ignoring other information. ‘Concentration’ captures the experience of sustained and determined application of will. The characterisation of its opposite as ‘distraction’ is a theme that will feature in the work presented later in this thesis.

At the time of James’s characterisation of attention, it was surmised that there were distinct memory subsystems within the human brain and that some of these were associated with attention. By the 1960’s neuroanatomical and psychological evidence was growing that attention was related to short-term or working memory which interacted with long-term memory systems and these were indeed anatomically separate subsystems within the brain.

An early conception of attention as selective filtering applied to information pathways that are bottom-up or in-line is due to Broadbent [1958]. He argued that all sensory information was processed in a rudimentary way and stored in an immediate or short-term memory. Further processing of this information to attend to salient features within it required access to a mechanism that was physically limited in capacity and thus required an attentional filter.

Broadbent theorised that top-down control processes guided this attentional process as it selected a channel or stream of information. This broad conceptualisation continues to underpin most models of executive attention. However, Broadbent had argued that information that fell outside the attended channels was not processed, beyond separating it out from the currently attended stream but Moray [1959] later showed that people performing an attended task could be influenced by semantic information from unrelated parallel tasks. The recognition that both selective and non-selective information processing occur in parallel is an important feature of subsequent models of attention-based behaviour.

In their extensive review of attention and identification, Lachter et al [2004] confirmed Broadbent's original proposition that higher cognitive processing of perception (i.e. identification of an object) cannot happen unless attention (and awareness) is used. Through the use of new experiments where the focus of attention was closely controlled they found that unattended to stimuli could not be identified and that *identification* priming of the of these unattended to stimuli was also not present

Since Broadbent's formulation of attention, several subsequent and often related accounts of attention have been developed. Perhaps most notable are those of Baddeley and Hitch [1974], and Norman and Shallice [1986] both of which contain a functional system which is assigned a supervisory role: initiating, monitoring and modulating higher-level processing and behaviour. Later models, such as those by Duncan [1993], Cowan [1999] and Baars [1997] draw appreciably on the work of Broadbent, Baddeley and Hitch and Norman Shallice. A recent model which takes a rather distinctive view is that of O'Regan and Noë [2001]. Each of these models is considered in turn.

3.2.1. *Baddeley and Hitch*

In developing their seminal model of working memory, Baddeley and Hitch [1974] posited a 'central executive' as an attentional controller. In contrast to Broadbent's emphasis on perceptual filtering, Baddeley and Hitch's executive is also concerned with integration of information and action selection. The three major components of the model are described below:

- A visuo-spatial sketchpad (also known as a scratchpad) holds and is capable of manipulating visuo-spatial information
- A phonological loop (also known as the articulatory loop) performs the same role as the visuo-spatial sketchpad but for speech based information

- The central executive performs the role of attentional control, co-ordinating the activities of the visuo-spatial sketchpad and the phonological loop. It also links these subsystems to long-term memory (not shown in *figure 3.1*)

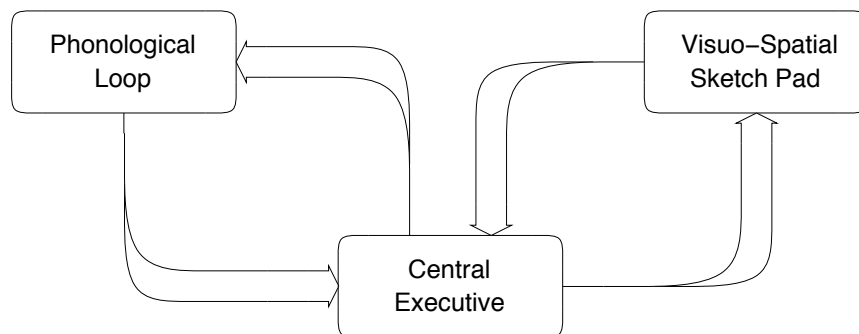


figure 3.1 The Baddeley & Hitch multi-component model of working memory, showing the controlling *central executive* and its two specialised slave processing systems, the *phonological loop* and the *visio-spatial sketchpad*

Within this model, there lies a distinction between attention as passive selective filtering and active executive control of attention. Some forms of selective attention are passive or involuntary. For example, early neural circuits in both the visual and auditory pathways select for features (such as edges or pitches, respectively) in ways that are not considered subject to will or even consciousness (as in the case of blind-sight). In contrast, Baddeley and Hitch consider the central executive to be an attentional controller concerned with active selective filtering and, beyond that, with integration of information processing and action selection. Specifically, Baddeley suggested that the executive maintained a representation of goals and intentions and that this may extend to representations of currently active behaviours.

3.2.2. Norman and Shallice

Evidence for the existence of an executive had been offered by Luria [1966; 1970] who analysed the organisation of behaviour and proposed that the brain was composed of major functional units that were both functionally and physically separable. One of these units spanned the frontal, pre-frontal and motor areas of the brain, which were involved in creating intentions and ‘programs’ (behaviours). Based in part on electroencephalogram (EEG) observations by W. G. Walter, these areas were shown to be active when a subject was paying attention and were quiet when attention was exhausted (or the subject was incapacitated). Injuries to this region severely affected attention and concentration.

In 1980, Norman and Shallice proposed a model for attention-based control of behaviour which drew on this presumption that cognition was controlled via a number of separate components [Luria, 1970; Allport et al., 1972; Baddeley and Hitch, 1974; Posner, 1978]. Norman and Shallice argued that a variety of processing units are used in action and cognition and that these can be classified as belonging to one of two distinctly different (and separable) systems, routine and non-routine as shown by *figure 3.2*.

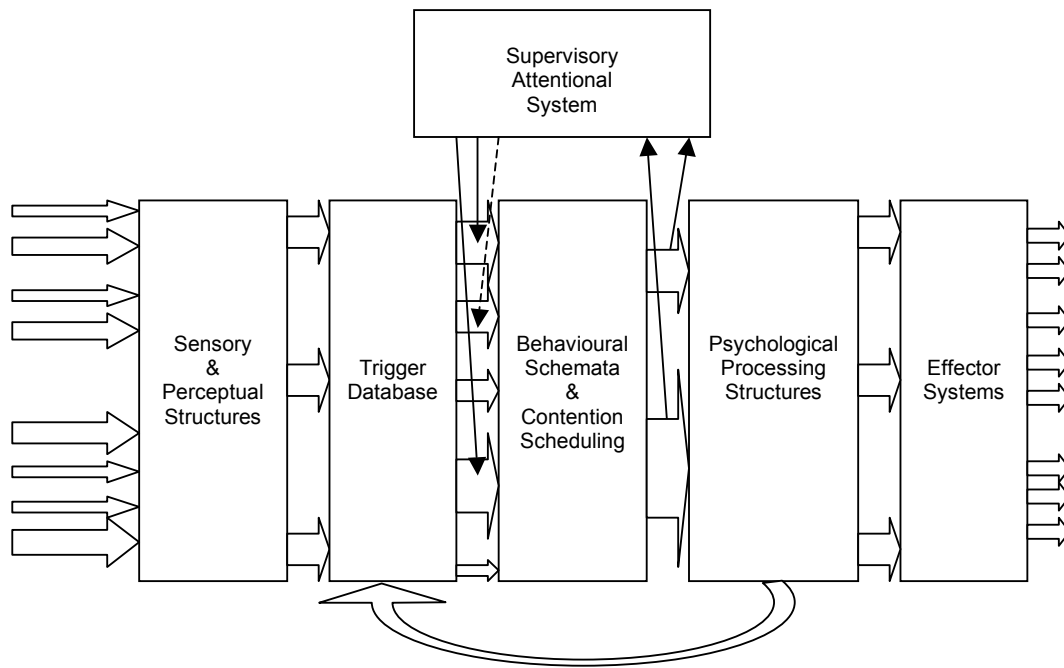


figure 3.2 The original Norman & Shallice model [1986] adapted from [Norman & Shallice, 1986] features a Supervisory Attentional System (SAS) which detects the expression of unwanted action (an interrupt mechanism) and attempts to assert willed action. Elaborated as a functional architecture, the Norman and Shallice model integrates both automatic (routine behaviour, modulated by the contention scheduler) and executive action selection (non-routine, modulated by the SAS) and comprises several functional sub-components

In the model (in which the language reflected computing terminology of the day) the system that guides routine behaviour was comprised of a perceptual system responsible for taking sensor information and presenting it to a ‘trigger database’. The trigger database mapped perceptual information to action or thought schemas (originally derived by Norman & Shallice from Piaget’s [1936] schemes). These schemas were believed to be ‘programmatically like’ structures, each one corresponding to a well learned but distinct thought or type of action. Schemas were connected to the actuator systems through special-purpose cognitive subsystems, which also took inputs from the perceptual system and fed information back into the trigger database.

One important property of the system was that multiple schemas could be active at any one time. This necessitated a conflict resolution or ‘contention scheduling’ system which prevented concurrent access to cognitive and/or motor subsystems by competing schemas (a function which Shallice speculated was associated with a region of the brain called the basal ganglia).

The non-routine system within the Norman & Shallice model incorporates a Supervisory Attentional System (SAS) which deals with non-routine action selection by modulating the routine selection of schemas by the contention scheduler. The SAS was presumed to be located within the pre-frontal cortex and to be active in the following situations:

- Situations that involve planning or willed decisions
- Situations where the behavioural response is not yet automatic (not well learned) or may contain unfamiliar or wholly novel action sequences
- Situations that involve correcting behavioural errors by ‘trouble-shooting’ already well learned behaviours
- Situations judged to be technically difficult or where the consequences of failure may result in injury
- Situations that require the suppression of a highly habitual but incorrect response

In a later paper Shallice & Burgess [1993] argued that the five situations above are actually subclasses of just two simple types:

- The contention scheduler generates an inappropriate response
- The situation has a high degree of novelty/unfamiliarity and hence no routine response is highly triggered

Whilst testing this new hypothesis in patients with damage to the prefrontal cortex (the SAS’s presumed location) Shallice & Burgess discovered that the SAS itself was composed of different subsystems. In a later paper they speculate on the fractionation of the SAS [Shallice & Burgess, 1998]. A simplified diagram is shown in *figure 3.3*.

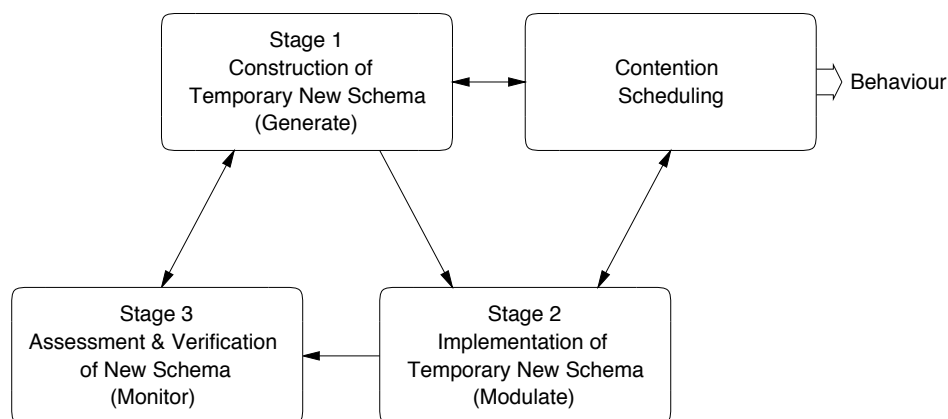


figure 3.3 A simplified diagram of the Shallice & Burgess Supervisory attentional system decomposition adapted from Shallice & Burgess [1998]. Box names in brackets (i.e. Generate) are the names used in this thesis and are not shown in the original diagram. Stage 1, *the construction of new schema* has several sub components that are not shown, but include problem orientation, goal setting, aspiration setting, strategy generation and episodic memory retrieval. It interacts directly with the contention scheduler and instructs the Stage 2 process, *Implementation of Temporary New Schema*, to build the temporary schema. Stage 3, the *Assessment & Verification of New Schema*, monitors the temporary schema set up by the Stage 2 process and verifies its operation and performance to the goals and aspiration set by Stage 1. If the schema is successful it is integrated into the currently operating schemas (not shown), otherwise a specific element of Stage 3 removes it

In the refined model the SAS is assigned a number of distinct sub-functions designed to deal with a range of degrees of novelty, extending from minor variation in an otherwise familiar context to total unfamiliarity. Shallice identifies them as follows:

- **SAS Monitor:** the SAS must be able to compare the currently expressed action with an intended action (as formulated by the SAS or other 'planning' units). The monitor may be thought of as an arousal, interrupt or warning mechanism that induces the activation of the other, deliberative, SAS sub-units
- **SAS Modulate:** when so aroused, the SAS must provide a modulatory signal that attenuates the salience of inappropriate tasks and potentiates the salience of appropriate tasks. (It is important to recognise that the modulation is a biasing mechanism, not a deterministic mechanism.) Shallice suggests three possible modulatory responses:
 - Attenuate the currently expressed behaviour for a given time and potentiate an intended behaviour
 - Attenuate the active behaviour for a given time and potentiate a 'default' or 'if all else fails' response
 - Attenuate all intended behaviours for a given time, allowing the contention scheduler to express a behaviour governed by perception of the environment alone
- **SAS Generate:** the SAS must create new strategies for solving novel problems

The issue of learning is left open in this model though there is an acknowledgement that there must be a means by which behaviour that is initially novel becomes more automatic and integrated into the system of schemata. Similarly, the role of affect in attention is not addressed in the original model.

3.2.3. Duncan

Related to the work of Broadbent, and of Norman and Shallice is Duncan's work on the role of goal selection in the control on behaviour [Duncan, 1993]. His studies of visual attention in primates and people have led him to hypothesise that attention is based upon filtering of perceptual input, achieved through the use of a limited capacity system [Broadbent, 1958] to which elements in perception have to compete for access. The competition is mediated by an associated weighting system that is generated (guided) by the matching of perceptual inputs to a template (an advance description of what is needed) relevant to the behaviours and goals of the agent. One of the primary roles of the limited capacity system is to make whole descriptions available for the setting of goals and the control of behaviour.

Although not explicitly described by Duncan as a subsystem, the central 'goal-weighting system' that performs the template weighting for goals (and therefore controls selection and behaviour) is presumed to be located in the pre-frontal cortex and is assumed to be a key system in the performance of general intelligence.

3.2.4. *Cowan*

Cowan's model of attention describes how a finite set of subsystems work to produce 'working memory' [Cowan, 1999]. This is markedly different to Baddeley's proposal for a single separable working memory subsystem. Cowan's Embedded Process Model comprises five basic principals of operation:

- Information in working memory comes from long term memory, the subset of currently activated long term memory, and the subset of activated memory that is in the focus of attention. It is presumed long term memory is itself hierarchically organised
- Subsystems that make up or feed working memory have process limits: attention is capacity limited but memory access is time limited
- Attention is controlled by a voluntary system (functionally, the central executive) and an involuntary system called the attentional orienting system
- Features of objects that remain the same over time can activate parts of memory however these features by themselves do not create awareness
- New memory and perceptual features are created by (conscious) awareness

Cowan's process based description of working memory distinguishes between attention and (conscious) awareness but argues that both are a function of working memory. Attention in this model, as with most of the models so far described, is regarded as the processing of some selected information whilst simultaneously ignoring other information. Well learned or automatic processing is considered less demanding of attention than novel or unfamiliar information.

A feature of the embedded process model is the possibility that features in memory can also be actively inhibited by attentional mechanisms. Inhibition of features in memory, directed by attention, appears to use the same resources as attention based processing in working memory [Engle, Conway, Tuholski and Shisler, 1995].

Baddeley's phonological loop and visio-spatial sketchpad appear to Cowan to be just two different kinds of memory activation coupled with specialist processes for reactivation e.g. visualisation. In Cowan's model the central executive regulates working memory but its internal functions or how it manipulates working memory are not defined.

3.2.5. *Baars*

In developing a theory of conscious perception, Baars' Global Workspace Theory (GWT) assigns a central role to executive attention [Baars, 1997; 2003]. Baars describes how the consciousness associated with willed action involves a selective attention system under dual control of frontal executive cortex and automatic interrupt control involving the brain stem, pain systems, and emotional centres [Baars and Franklin, 2003]. This distinction between the dual attention systems is of significance here; the interrupt system invites a deliberative response from the executive system.

Willed attention to action selection may be transient (attentional effort is exerted momentarily), intermittent (attention is exerted periodically) or sustained (attention is constantly applied) and to accommodate this, Baars suggests that once expressed, a willed response is unconsciously monitored.

The definitive feature of GWT is the proposition that multiple, distributed specialist processors, operating in parallel, compete for access to a limited-capacity global workspace which selects a winning coalition of such processors and then broadcasts associated information back to the set of all processors. This cyclical flow of information produces an experiential sequence in which there is a conscious decision to attend followed by unconscious attentional activity (monitoring), which produces targeted conscious contents.

3.2.6. *O'Regan and Noë*

A somewhat different view of the attentional mechanism is taken by O'Regan and Noë [2000; 2001]. In their account, the effects of attention are brought about not by top-down modulation of a mental representation of the world, but by control, both willed and automatic, of exploratory activity in relation to the world itself. The activity involves application of the perceiver's (acquired) knowledge of the sensory changes that are produced by motor actions, so-called sensorimotor contingencies. For example, the sensorimotor contingencies governing vision involve movements of the eye, altering the point of view; those of audition involve movement of the head, etc. Subsets of sensorimotor contingencies are associated with specific sensory effects, revealing specific sensory information about the object or scene being produced, e.g. shape, colour, orientation, location in relation to other objects, etc.

O'Regan and Noë share the view of attention as bringing to awareness, in their case, by an integration of a mastery of sensorimotor contingencies for the purposes of thought or planning action. Shifting attention involves switching between one subset of sensorimotor contingencies and another. Mastery of sensorimotor contingencies implies a high level of automaticity in their expression (i.e., there is little awareness of their application), but in novel contexts there may be a requirement for sustained, willed effort to secure the desired sensory information. A distinctive feature of this model is that there is no explicit mental representation of the world that is subject to processing, manipulation, or reasoning; it captures the notion that the world is its own representation.

3.2.7. *Selecting a Model for Practical Research*

In looking to select a theoretical model of executive attention as the basis for subsequent implementation and experimentation, a significant factor in favour of the Norman and Shallice model was that Baddeley had himself concluded that only the Norman & Shallice model matched the properties of information integration and control of action [1993]. Subsequently, this view was echoed by Gathercole [1994].

Beyond these significant endorsements of the Norman and Shallice model, the following observations were made:

- Both the Baddeley and Hitch and the Norman and Shallice models have clear analogues of episodic and procedural memory and the recurrent path between them
- Neither the Duncan nor Cowan models significantly extended or refined the Norman and Shallice model although between them they did give a more appropriate emphasis to the attentional control operating over a limited capacity system which represented the most active parts of memory and current goals and intentions
- In comparison to Baars's GWT model, the Norman and Shallice model features a clearer architectural separation between the mechanisms for automatic and willed action selection and that this separation was a better reflection of the functional specialisation across the vertebrate brain
- In respect of O'Regan and Noë, their model of willed and automatic action suggested a treatment of sensorimotor contingencies as a subset of the motor actions over which there is executive attentional control
- The model of Burgess and Shallice provides a useful refinement of the SAS functions
- None of the models deals explicitly with the issue of learning. (This is investigated below.)

The above considerations led to selection of the Norman and Shallice model as the initial basis for implementing a functional architecture.

3.3. Attention-based Learning

Executive attention as described above is concerned with problem solving; it is invoked when there is a need to adapt to a novel or relatively unfamiliar situation. The degree of novelty may be large or small. However, adaptivity is not learning per se. An adaptive agent can deal adequately with the contingencies of a dynamic environment as it pursues its goal(s) but, if it does not also learn, it will always respond to a given contingency in the same adaptive way. In a dynamic and unpredictable environment where there is always a degree of novelty and sources of distraction it will not improve its goal-related performance over time.

In respect of learning, and in particular in the context of neural systems, an important distinction is between machines systems that undergo continuous learning and those for which there is a distinct learning or 'training' phase prior to an 'execution' phase. A second important distinction is between systems that learn on the basis of feedback from an external entity (a 'teacher') and those for which the feedback is autonomously derived on the basis of an intrinsic mechanism. Maes [1994, pp. 151-2] considers that a defining characteristic of autonomous systems is that they learn continuously and do not rely exclusively on extrinsically generated feedback.

A common approach to deriving a continuous and intrinsic learning signal is to invoke ‘drives’ that reflect internal states such as attraction/repulsion, hunger/satiation, etc. [Hull, 1943; McFarland and Sibly, 1972; McFarland, 1985]. Individual drives can be enhanced or suppressed by experience and the next action is chosen by simply selecting the behaviour (and its associated set of actions) with the highest strength drive.

In the context of the research approach adopted in this thesis, one objection to such devices is that they substitute for complex physiological and psychological systems that themselves warrant modelling. Even at a pragmatic level there are several major issues with a drive based architecture [Tyrrell, 1992; 1993b]: defining the behaviours is problematic, learning is undefined, as is how to combine different classes of perception, and the system may respond inappropriately if several drives are equally salient for a particular stimulus pattern.

3.3.1. Deriving an Attentional Mechanism for Learning

In early experiments on learning in autonomous machines, Walter [1953] noted that:

- Learning begins with failure (implying the existence of a goal)
- Successful learning is reinforced by repetition

In the context of executive attention, these ‘discovered principles’ suggested that the magnitude of attentional effort was a neuropsychologically plausible intrinsic learning signal. (Exploring this possibility became an important objective of this research programme).

One possible route by which attentional activity may contribute to learning is via the hippocampus. Normal functioning of the hippocampus depends upon attention and interactions with other elements of the limbic system, a group of brain structures (hippocampus, amygdala, fornicate gyrus, archicortex, hypothalamus) associated with affective responses (emotions mediated by the endocrine system). Shastri [2002] reviews the evidence suggesting that learning of new reactive behavioural associations (episodic learning) may be mediated by the hippocampus. Whilst there is widespread agreement about the role of the hippocampus in episodic learning, there is currently no consensus as to the neural mechanisms at work. The attentionally derived learning signal which features in the research presented in this thesis must therefore be considered as conjecture.

3.4. Related Research: Neurally Implemented Models of Attention

Whilst much of the literature considered in this section predates the framing of the research programme, some of it is contemporary with the work done. This latter work not only served to affirm the continued relevance of the research programme but also to confirm developing perspectives, design decisions and interpretations of experimental results.

For researchers exploring attention-based behaviour within the context of neural computing (connectionism/PDP), the overwhelming majority of the work undertaken has focussed upon automatic (reactive) behaviour. Considerably less work has been undertaken on willed behaviour.

It is possible to identify a sequence of studies of attention-driven task-related behaviour that can be traced back to an early experiment by Cohen, Dunbar and McClelland [Cohen et al., 1990]. More recently, there has been renewed interest in neurally-based attention, often motivated by research into consciousness.

3.4.1. *Cohen, Dunbar and McClelland*

In their early paper on attentional control of task selection, Cohen, Dunbar and McClelland [1990] developed an architecture which allowed them to examine the so-called Stroop effect. When humans switch between cognitively distinct tasks there is a perceptible delay in task execution. The classic Stroop experiment presents a subject with words representing colours (red, blue, green, etc.) written in different coloured inks; the word and the ink in which it is written may or may not be the same (e.g. the word red, may be appear in red, green or blue inks). The subject is instructed to utter either the word or the ink colour when presented with a subsequent stimulus. Whilst the task remains unchanged over a sequence of stimuli, responses to each new stimulus are relatively fast; however, whenever the subject is required to switch between tasks, there is a very marked delay in responding until they settle to the new task.

The network developed by Cohen, Dunbar and McClelland to perform this task dealt with two word-colour alternatives (red, green), featured a winner-takes-all response layer, and the attentional or willed input was provided by a ‘context layer’ providing a sustained exogenous signal determined by the experimenter. The context layer encoded which of the two tasks was to be performed at any time. Switching the pattern of activation in the context layer caused the network to change task. This system was later modified to incorporate continuous as opposed to discrete processing and was adapted to perform other task switching exercises.

3.4.2. *Gilbert and Shallice*

Gilbert and Shallice [2001] developed a related model of attentional control. This model dealt with three word-colour alternatives, and modified the architecture so that there were two separate networks, each trained to perform one of the two tasks, integrated with a task demand layer, serving a similar function to the context layer of Cohen, Dunbar and McClelland. Again, the task demand layer was controlled by a top-down exogenous control input determined by the experimenter.

We can consider these related models in relation to the three features of attentional systems sought by LaBerge [1999] (described earlier). The issue of whether the networks considered so far exhibit willed selection of a less salient task over a more salient task is not entirely straightforward. In this experimental paradigm, task salience cannot be derived from the stimulus itself; when presented with

the word red written in green ink, there is nothing that indicates whether the response should be utterance of the word 'red' or the word 'green'. Task salience is governed entirely by an (exogenously derived) attentional signal encoded in a task specification layer. If one takes the view that salience derives from the stimuli embedded in the environment, then these models do not exhibit this feature. However, if one accepts that salience is simply the current level of task activity, however derived, then the fact that in these networks task salience is determined wholly by the bias provided by the task specification layer is of no account and the desired feature is present.

In all cases, the maintenance of the current task is governed by a persistent attentional signal. Gilbert and Shallice [2001], acknowledge that this represents a departure from standard attentional theory, which holds that attentional effort in task switching is selective and intermittent.

The property of preparation (or priming) requires that expression of one task readies the subject to express a different but related task. None of the networks considered featured such a property. None of the models use memory to sustain task focus, unless one includes the persistent expression of current intention by the task specification as memory.

In respect of the functionality required of a SAS, these networks demonstrably apply modulation of information flow to focus and sustain attention to task. However, by virtue of the experimental paradigm, there was no need for an autonomous monitoring function (it is only needed when attention is not persistent), nor for a capacity to generate and learn responses to novel problems, and so these features are understandably absent.

3.4.3. *Cooper*

Cooper [2003] has augmented an earlier system for managing routine action selection [Cooper and Shallice, 1997; 2000] to include intrinsic supervisory processes, including some monitoring and error recovery, which do not depend upon exogenous control. The system is capable of generating sequences of basic actions associated with the high-level task (preparing a lunch box selection). In the model, each task is realised as a hierarchy of sub-tasks. Each task in the hierarchy is assigned pre- and post-conditions whose truth values will depend on the current state of the environment. Having introduced pre- and post-conditions for tasks and sub-tasks, Cooper is able to implement a monitoring and error correction system. Once each selected sub-task of a higher task has been attempted, its post-condition is evaluated. If false, the task is deemed incomplete and failed sub-tasks continue to receive selective excitation from their parent task schema.

As before we can consider this model in terms of the features sought by LaBerge [1999]. The model does select between sub-tasks of differing salience, where the use of task preconditions allows salience to be driven by the state of the environment. There does not appear to be an explicit preparation of priming mechanism in which a currently active sub-task raises the excitation of the next task in sequence. However, the fact that the existence of noise in the signals passing between task nodes

appears to lead to task skipping, in which an ensuing task is invoked prematurely, suggests that priming may be implicit in the implementation. Attentional persistence is realised through the error detection system which maintains the activity of a task whose post-condition has not been satisfied. Cooper [2003, p. 52] describes two conceptual approaches to ordering of sub-task expression. The first approach (which appears to have been implemented) involves a task selectively exciting some of its own sub-tasks. The second approach (which may have been adopted, though this is not clear from the paper) would involve external inhibition of tasks whose pre-conditions are false or whose post-condition is true. As Cooper observes, the former approach constitutes automatic control; the latter approach would constitute attentional task selection.

3.4.4. *Taylor, Fragopanagos and Kasderidis*

Taylor has developed the CODAM model of attentional control, based upon engineering control theory [Taylor, 2003; Taylor, 2004]. The illustrative domain involves control of motor function in response to a visual stimulus. The CODAM model features an inverse model control (IMC), which modulates activity in analogues of early sensory and motor cortices. The IMC receives inputs from a goal module, which encodes the action to be performed and provides a biasing attentional signal as well as processing; and from a monitor which calculates any error between the goal and either the actual or the predicted sensory input (considered as distinct working memory buffers). The monitor gives rise to an attention-based signal that is used by the IMC to learn less error-prone motor responses.

Taylor presents the results of three simulation experiments which lend weight to aspects of the CODAM model. First, he demonstrates at least qualitative agreement of temporal information flow (understood as relative timing of attention-related potentials) between actual EEG signals and information flow in the simulation. Secondly, he reproduces a well known lesion study (simultaneous extinction) to illustrate that attentional control of the sensory and motor cortices is distributed (right and left hemispheres respectively). Finally, he demonstrates that the model reproduces the attentional blink in which the ability to respond to distinct visual targets which require an attentional shift of focus is diminished as the delay between occurrence of the related stimuli approaches the time needed to process the first image and prepare for the onset of the second.

Fragopanagos and Taylor [2004] describe an application of a closely related control model, investigating the behaviour of the model in relation to two further standard explorations of human attention. The control model is similar to the CODAM model described above, except that the monitor compares the motor output with the visual input and does not draw upon a predicted sensory input state. In the first simulation, the controller, responding to an error signal from the monitor, learns which of two motor responses (analogous to finger movements) to make in response to visual stimuli. In the second simulation, the IMC controllers of visual and motor cortices respectively are lesioned. The results of both simulations accord well with results from human subjects; the second simulation serving again to suggest distribution of the controllers for vision and motor function.

Kasderidis and Taylor [2003] use an extended attentional architecture to control a simulated robot. In this instance, the monitor compares the expected sensory state of the robot with the actual sensory state, producing an attentional arousal signal (called the attentional index) which is proportional to the error detected. The arousal signal is used both as a learning signal to the controller so that it learns motor responses and as input to a process that evaluates competing actions. The elaboration of the earlier control model relates to the representation of goals as goal hierarchies, with high level goals being decomposed to lower level goals. At any level, the goals compete for expression through mutual inhibition. The simulated robot has three sensors (power, proximity of other objects, and position/speed) and one effector capable of moving the robot one step at a time in a discrete Cartesian grid. The robot has a high level goal: 'transport an object from A to B whilst avoiding collision'. This goal decomposes to a 'transport' sub-goal and an 'avoid collision' sub-goal. In turn, 'transport' decomposes to four sub-goals: 'goto A', 'pick item up', 'goto B', 'drop item'. At the next level of decomposition, 'goto x' reduces to the goals 'plan route' and 'move', respectively. The 'plan route' action determines the next move based upon predictive assessment of the movements of other objects in the environment. If these predictions are in error, an attentional signal (collision avoidance index) is generated, the signal being inversely proportional to the proximity of the potential obstacle. An action index, which incorporates the attentional and collision avoidance indices, is calculated and if a decision threshold is exceeded the collision avoidance behaviour is selected preferentially over the transport behaviour. The collision avoidance index is used as a (back-propagation) learning signal for a neural network that predicts the movement of the threatening obstacle. The model is used in a simulation that compares the relative performance of the robot as described, compared to two grossly lesioned robots: the first lesioned to remove learning; and the second lesioned to remove both learning and prediction.

Taylor and Fragopanagos [2003; 2004] have extended the CODAM control model to incorporate the effect of emotion on attention. In humans, the attentional blink (described above) is attenuated by the emotional content of the second stimulus (e.g. the effect is diminished if one's own name or a familiar face is the stimulus to be recognised). In patients with a lesion to the amygdala, there is no attenuation. By incorporating an emotional signal comparable to that generated by the amygdala (part of the limbic system) into a simulation, and using lesion studies, they are able to reproduce this phenomenon.

Taken collectively, the applications of Taylor's CODAM model described above exhibit the properties associated with executive attention, along with the requirement for learning of automatic responses. The goal module serves as a memory to maintain task focus. It receives directly a representation of the input from which CODAM determines which of a number of attentional goals apply at any instant. Thus task salience is derived entirely from the environment. In this sense the term 'goal' is synonymous with 'currently specified task'. The concept of goal is more clearly present in [Kasderidis and Taylor, 2004] where willed control is exercised over more than one possible task (e.g., suppression of the transport task in order to avoid collisions). There is task-related priming of anticipated actions in [Kasderidis and Taylor, 2004] and there is error-driven learning to reduce the occurrence of errors. It is

not clear whether the input from the goal module to the IMC is intermittent or sustained, it appears to be the latter.

3.4.5. *Franklin, Shanahan and Baars*

There are two independent applications of Baars' Global Workspace Theory (GWT) [1997]: The first due to Franklin et al. [2005] and the second to Shanahan [2005].

Franklin's IDA model [Franklin et al., 2005] can be viewed as an implementation of Baddeley's working memory using Baars' concept of Global Workspace Theory (GWT). IDA is a multi-agent system [Franklin and Gresser, 2001], an early implementation of which was able to billet US Navy personnel, responding to (electronic) notices of vacancies and requests for postings by assigning staff appropriately. IDA can communicate in natural language via email, and the application solves a nontrivial instance of constraint satisfaction.

Although Franklin's implementation of GWT in IDA contains no neural networks, it does incorporate some connectionist elements, (e.g. software agents exhibit 'levels of activity' which are used in computation by other agents and thus determining the flow of information within the agent network [Franklin and McCauley, 2002]) justifying its inclusion here.

At the heart of Franklin's IDA is a non hierarchical action selection mechanism dubbed a 'spreading activation network' an implementation of a proposal by Maes [1989, 1990, 1991]. Nodes representing actions and motivations and are made up of the following:

- Preconditions which much be true for the node to be considered for execution (e.g. for a robot; water is available, thirst)
- An 'add list' containing conditions that executing the node will probably make true (e.g. water may still be available)
- A 'delete list' containing conditions that executing the node will probably make false (e.g. thirst after drinking)
- An activation level, reset if the node gets executed
- The actual code to be run, which for actions can be a generated pattern that is relayed to the motors

At each time-step the activation levels are re-calculated across the network, the node with the highest activation level getting executed (resetting its activation level to zero for the next time-step). The various external, inhibitory and excitatory links from node to node form relationships between nodes, goals, motors and sensors.

In the most recent elaboration of the IDA model, which extends the working implementation described above to include some features which are not yet implemented, Franklin et al. [2005] present a nine-

step cycle of computation that makes concrete the cycle of computation implicit in GWT. Sensory stimuli are ‘chunked’ to form current ‘percepts’. These are combined with representations of recent percepts from transient episodic memory and from long term memory. Event specific attention ‘codelets’ seek to build a coalition of associated information codelets on the basis of the amount of relevant information content found in the current internal representation of the environment. A winning coalition is selected and its contents broadcast. Behaviour codelets respond depending upon their relevance to the information broadcast. Some behaviour codelets initiate a behaviour stream, if one is not in place. Competition between behaviours is resolved by selection of a single behaviour for execution. Execution of the behaviour codelet results in the creation of an expectation codelet (a type of attention codelet), which competes for attention in the ensuing cycle(s) if the anticipated results of the behaviour are not manifest in the perceived environment.

Shanahan [2005] has developed a neural implementation of Baars’ GWT model in which attention is implicit in information flow through the architecture. In the context of robot action selection, Shanahan’s action selection architecture features a first order, purely reactive, system, modulated by a higher order system. The first order system evaluates (assigns salience to) sensory input to determine, in parallel, alternative courses of action. In response to the same sensory input and the currently selected action, the higher order system predicts, off-line and in parallel, a number of trajectories through the robots sensory motor space. The outcomes of these parallel rehearsals inform an affective assessment of the currently selected action based upon the extent to which it leads to a rewarding or punishing outcome. This affective assessment provides a signal which modulates the salience of the selected action, either reinforcing its expression or causing it to be suppressed in favour of a more rewarding alternative.

Clearly, IDA selects less salient over more salient tasks through a number of biasing mechanisms, e.g., agents representing affective states, timekeeping agents, etc. In IDA, the affective content of external and internal representations is recognised during perception (early stages of the cycle) and thus informs the unfolding of the information flow in later stages and cycles [Franklin and McCauley, 2002]. Priming is also implemented, as described above. The operation of attention, in respect of any one task, is intermittent. And the frequency with which a task is revisited is governed by the dynamics of the competition between attention codelets (with their coalition of information agents) and the rates at which the activity levels of the codelets decays in episodic memory. From the perspective of SAS functionality, IDA has a monitoring mechanism (e.g., expectation codelets). The model stipulates a role for a planning mechanism to tackle non-routine problems but the mechanism for this is unpublished [Franklin et al, 2005]. One form of learning in IDA uses internal reinforcement mechanisms, the details of which depend upon the target memory subsystem. The reinforcement signal is derived from selection of an attention or behaviour codelet so that frequently attended perceptual or behavioural associations are sustained whilst others are allowed to decay. The model specifies a mechanism to enhance the automaticity of unfamiliar skills.

Shanahan's model and its implementation exhibit a mechanism by which action salience is modulated in accordance with the outcome of higher-order processing, and the architecture closely reflects Norman and Shallice's separation of the pathways for automatic and higher-order action. There is no explicit representation of will or intention to form the basis of willed action selection. Rather, in so far as there is a will, it is implicit in the architecture and is introduced through the affective biasing of action selection: 'prefer actions which are relatively the most rewarding'. A distinctive element of the model is its ability to use imagined future outcomes to evaluate courses of action. Shanahan uses forward association from the current input, as opposed to backward association, which would demand the representation of some goal state [Garagnani et al., 2002].

3.4.6. Relationship of Previous Work to the Research Conducted

It is appropriate to indicate the similarities and differences between the work described above and the work of the research programme which is the subject of the following chapters.

In comparison to the early work of Cohen, Dunbar, McClelland, Gilbert and Shallice, there is a significant difference in the scale of the system and the functional refinement of its subsystems and their integration. The persistent, extrinsically controlled attentional control signal is replaced by a more plausible intermittent and autonomously generated signal.

In relation to the model of Cooper, the model developed here echoes the hierarchical organisation of tasks and the priming of lower tasks by higher tasks, but develops an independent neural implementation of a task hierarchy as one of the functional subsystems in the overall control model. Cooper's attentional system is embedded in the task hierarchy itself. In contrast, the model used here separates the attentional system (reflecting the apparent separation seen in the brain) and the task hierarchy.

Considered as a whole, the work of Taylor et al., which was conducted contemporaneously but independently of the research described in this thesis, has many more similarities than other work. It shares the notion of functional separation of perception and action, the attentional signal is intermittent, there is a stored representation of a goal, and there is an attentionally derived learning signal. Lesion studies are used to explore the properties of the systems developed. The robot developed to evaluate the model of Kasderidis is assigned what is essentially the same goal.

One significant difference occurs in the detailed design of the control architecture, especially in relation to the structure and function of all subsystems and in the implementation of contention scheduling and of working memory. A second significant difference is in the derivation of the learning signal and its relation to the rest of the system. Taylor et al. consistently use the 'monitor' function of the attention system to derive a learning signal which is proportional to the difference between the expected state of the system and the actual (sensor driven) state of the system. The approach taken below is to use the modulation function as the basis for the learning signal. Both are conjectural (see

above) and with current knowledge are best considered as equally plausible alternatives and may even operate together. In implementing attentional learning, Taylor et al. do not use a pathway that involves the affective system. The implementation of attentional modulation by Taylor et al. means that lesions to the system tend to be large scale (this seems especially so of Kasderidis and Taylor, where the first lesion eliminates all learning and the second all learning and prediction. The development of the attentional system considered in the research presented below allows for much more fine grained lesioning so that more subtle (and significant) effects can be explored.

Echoing the reason for not adopting the Baars model of attention, the model featured hereafter gives a clearer architectural separation between the mechanisms for automatic and willed action selection that is a better reflection of the functional specialisation in the brain.

Finally, the major distinction in respect of the work of Shanahan is that Shanahan's means of prediction uses an off-line second order system to evaluate alternative courses of action. In contrast, the model developed here uses priming within the reactive system to provide a forward model that is capable of considering a number of possible (likely) future scenarios. The two approaches are not contradictory and one interpretation of Shanahan's model is that it could be considered as an implementation of the SAS Generate function specified by Shallice.

A secondary distinction with Shanahan's work is that rather than having an explicitly stored goal, his system has an implicit goal in a drive, represented as the reward-seeking preference of the affective system which decides which of the future courses of action to pursue.

3.5. *Summary*

This chapter has reviewed the literature that framed the research programme presented in the succeeding chapters. It examined a range of neuropsychological models of executive attention and identified Norman and Shallice's functional model as the basis upon which to design and implement a working system. It examined the evidence for an intrinsic learning signal based upon attentional activity and incorporated a research objective to explore how such a signal might function. Finally it considered related work by other researchers and pointed to relevant similarities and differences.

Chapter 4

4. Model Design

This chapter develops a functional account of a neural architecture for executive attention (referred to as the *Attentional Architecture*). The functional production rule architecture of Norman and Shallice's model of a Supervisory Attentional System (SAS) [Norman and Shallice, 1986; Shallice, 1988] is used as a starting point, with the chapter having three main parts:

- The first part provides an initial overview of the neural Attentional Architecture and its subsystems
- The second part presents a more detailed account of the main functional subsystems and sub-subsystems. It is prefaced by descriptions of two widely distributed subsystems which play a fundamental role in mediating information flow between the main functional subsystems within the brain and which have been incorporated in the architecture presented here. The first subsystem can be considered as providing an interface between the behaviour control system and the sensors and actuators. The second subsystem serves to resolve competition for expression (internal or external) by competing subsystems and provides the equivalent of the contention scheduling mechanism described by Norman and Shallice. The major memory subsystems are then considered in turn: episodic memory, procedural memory, working memory; followed by an account of the Supervisory Attentional System and the affective system, including how adaptive learning is achieved. Learning mechanisms differ between subsystems and could have been considered as each subsystem is presented. However, attention-based learning is best understood at an architectural level by taking a view of how the subsystems interact to achieve learning. Hence, the functional account of learning is presented after other elements of the Attentional Architecture have been considered
- At the end of this chapter, a high-level summary of the Attentional Architecture is presented as a series of annotated diagrams. These provide a 'reference document' against which the subsequently implemented system can be compared

4.1. An Overview of the Attentional Architecture

Building upon the functional architecture of Norman and Shallice's original production rule system (considered in *Chapter 3* and reproduced graphically in *figure 4.1*) this chapter develops a corresponding neural model at a more detailed level of functional design. This model extends all of the previous models of attention-based task selection developed within the neural/PDP paradigm as described in *Chapter 3*, and previous models developed as part of this research programme [Garforth et al., 2003, 2004]. It does so in at least three respects:

- This architecture (and the subsequent implemented system) exhibits a scale of functional integration for attentional control of action selection that has not been demonstrated previously
- The detailed treatment and exploration of executive attention using lesion studies is novel, particularly in neural systems
- The introduction of an intrinsic attention-based learning signal that allows initially novel behaviours demanding of executive attention to become increasingly automatic and for future task related distractions to be ignored, is also novel

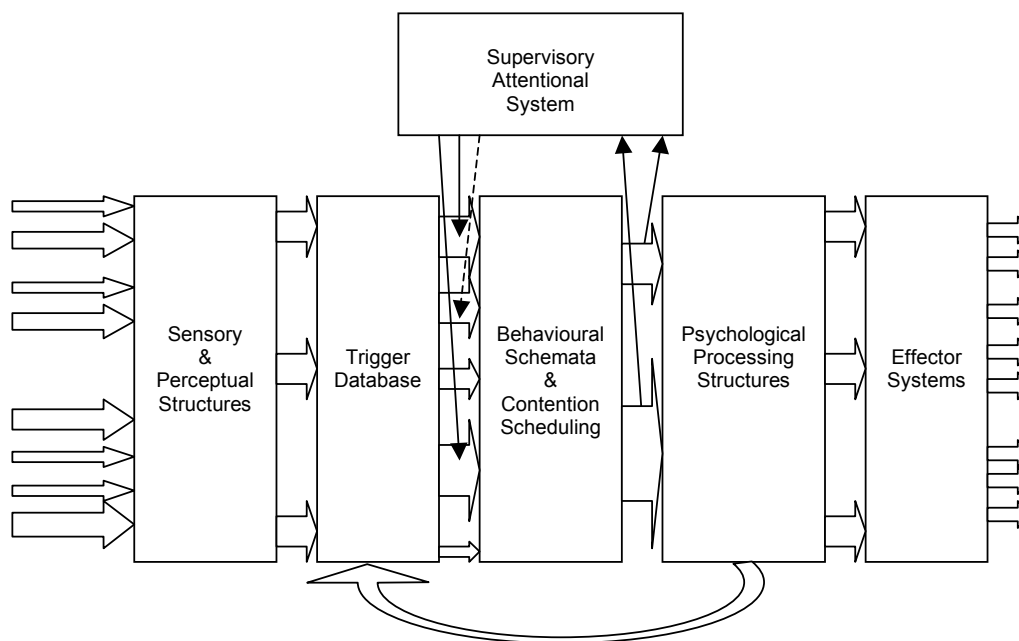


figure 4.1 The original Norman & Shallice architecture for executive control of routine and non-routine behaviour (after [Norman & Shallice, 1986]). Information from sensory pathways is mapped by a trigger database into a collection of behaviours (schemata); competing (contradictory) behaviours are subject to selection by contention scheduling; the output of psychological processing systems is mapped onto effector systems and provides feedback to the trigger database, reinforcing persistence of expression. The Supervisory Attentional System (SAS) modulates the triggering signals (solid arrows indicate potentiation, broken arrows indicate attenuation). The SAS monitors the selected behaviours, producing an 'warning' signal if there is a separation between intended and expressed action and subsequently applying a modulatory signal to contending behaviours

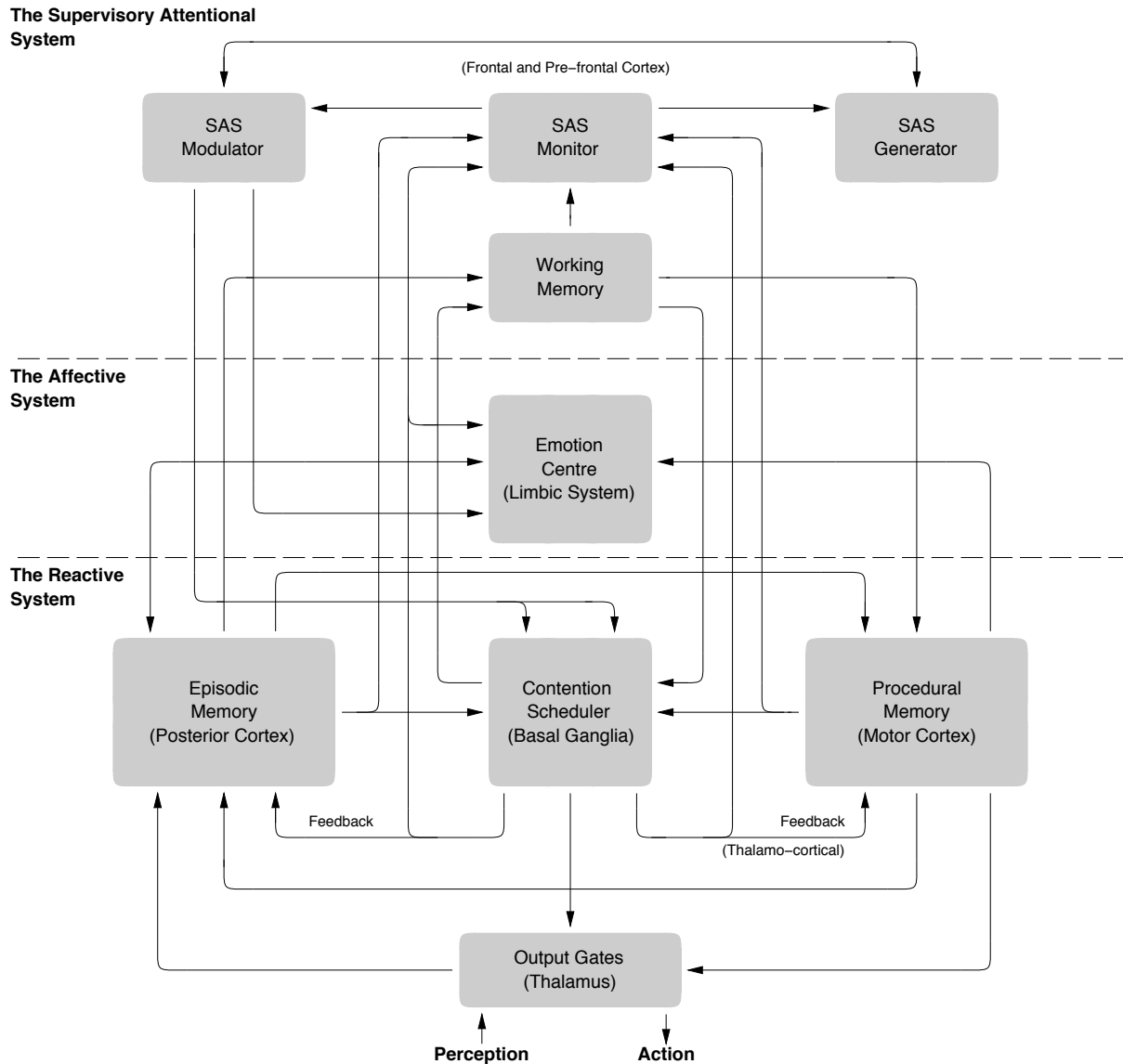


figure 4.2 An overview of the proposed architecture for executive control of behaviour showing the reactive system consisting of episodic and procedural memory giving direct coupling between perception and action, the affective system, and the Supervisory Attentional System (SAS) with its sub-components: Monitor, Modulator and Generator. Arrows indicate major information pathways (where functions are closely associated with anatomical structures these have been indicated).

The Attentional Architecture, has three major integrated subsystems (see figure 4.2 for an overview):

- A **Reactive System**, containing hierarchically organised episodic and procedural memory structures that together with the contention scheduler (not shown in figure 4.2), provide for automatic (instinctive or well-learned) behaviour
- The **Supervisory Attentional System** (also known as the *executive*), including working memory. This provides executive functions for autonomously detecting circumstances in which attentional effort is appropriate, adapting and modulating 'attended to' behaviours to express willed behaviour.

- An **Affective System** that mediates acquisition and integration of initially willed (attention driven) behaviours by the reactive systems under the influence of the Supervisory Attentional System

At this highest-level of functional description Norman & Shallice's architecture has been reinterpreted as a large-scale, modular, neural network in which established neuropsychological and neuroanatomical relationships at the functional-structural level has been maintained. In doing so, this architecture elaborates and incorporates additional subsystems; specifically

- Subsystems for hierarchically structured [Baerends, 1976; Tyrell, 1993], episodic [Fuster, 1995; Baddeley et al, 2002], procedural [Fuster, 1995; Tulving, 1993] and working memory [Baddeley & Logie, 1999; Cowan, 1999; Fuster, 2003]
- An attentional filter [Broadbent, 1958; Lachter et al, 2004], created by limiting access to working memory from episodic memory
- Priming in episodic memory using salience inputs from procedural memory [Frith, 1998], enabling the architecture to anticipate the consequences of its actions (including actions it may not actually perform)
- An affective system to simulate functional elements of the limbic system [LeDoux and Fellous, 1995, p. 356] and specifically the hippocampus [Shapiro & Eichenbaum, 1997; Fuster, 1995] which is responsible for neuromodulators which influence learning (and provide a link to emotion) through long term potentiation (LTP) in memory subsystems. These modulate connection strengths [Brown & Chattaraji, 1995, p. 454] in episodic and procedural memory promoting automaticity of frequently invoked behaviours to reduce attentional burden
- Gated information pathways and contention scheduling using a model of the basal ganglia [Alexander, 1999; Prescott et al, 1999] and the thalamus [Mumford, 1999; Gurney, 2001] providing perception to action binding

The following sections expand upon the description of the functional subsystems above, and their sub-components in turn, pointing to architectural issues that reflect functional requirements and known neuroanatomical structures. However, as indicated above this more detailed account is prefaced by a section which deals with the interfacing of the control architecture with sensors and effectors, and a section dealing with contention scheduling.

4.2. Major Functional Subsystems

The structure of this section reflects the overview above and will consider the elements of the reactive system, the supervisory attention system including working memory, and the affective system in turn. Whenever appropriate, reference is made to supporting neuroanatomical evidence, but in many instances this knowledge is only available at the macro architectural level, and in many cases such evidence is contested or even absent.

4.2.1. The Reactive System

The account of the reactive subsystem begins with an account of the subsystem providing functions associated with sensor input and control output, as well as a number of basic information pathways. This is followed by an account of the subsystem that provides the function Norman and Shallice describe as contention scheduling. Both these structures support information flow between the main memory subsystems supporting reactive behaviour: episodic and procedural memory.

4.2.1.1. Reactive Sensors, Effectors and Perceptual Structures

Sensor and effector (motor) systems are connected together in a number of ways. Anatomically, reflex ganglia and afferent and efferent neurons form a spinal column that routes information from sensors and to effectors via the cortex through ‘gateways’ in an anatomical structure called the thalamus [Mumford, 1999; Gurney, 2001]. The thalamus is thought to provide a limited form of perceptual attention by modulating sensory information pathways. It inhibits (motor) output until ‘disinhibited’ by signals from the basal ganglia (see *section 4.2.1.2*). The thalamus is also the medium of thalamo-cortical feedback which reinforces active segments of memory and also provides recurrent connections from the cortex and pre-frontal cortex, the site of (some) executive functions.

In the Attentional Architecture a structure functionally equivalent to this description of the thalamus provides input from sensors into the lowest levels of episodic memory as well as providing output for the lowest levels of procedural memory. It provides a feedback loop from procedural to episodic memory which promotes ‘persistence’ [Macfarland, 1989] or ‘perseverance’ [Snaith & Holland, 1990] of currently expressed behavior so that minor fluctuations in perceptual experience do not result in rapid behaviour switching. It also provides recurrent connections between hierarchically organised layers of episodic and procedural memory (see *section 4.2.1.3*).

4.2.1.2. Conflict Resolution and the Contention Scheduler

In respect of automatic action selection, ‘contention scheduling’ is a mechanism that allows all salient behaviours to be expressed at the same instant as long as there is no *contradictory* demand placed upon any effector. Contention scheduling is associated with the basal ganglia [Norman & Shallice, 1986; Alexander, 1999; Prescott et al, 1999]. The basal ganglia, comprises several somatotrophically organised, layered, parallel systems with local lateral excitation and inhibition which serve to prevent incompatible and incoherent access to information pathways and motor systems by competing systems.

Behaviours selected for expression by the contention scheduler are routed through output gates in the thalamus to effector systems. By default, the gating system inhibits the expression of salient behaviours (directly from procedural memory), the contention scheduler actively disinhibiting those behaviours selected for expression.

Whilst it is known that the basal ganglia is used primarily by procedural memory (including the pre-motor cortex and pattern generators in motor cortex) it physically extends to associative areas within the cortex and it is used within the Attentional Architecture to select between contending information pathways seeking expression, such as those from episodic memory into working memory, as considered below.

4.2.1.3. Episodic and Procedural Memory

The close coupling of episodic and procedural memory subsystems is the basis of reactive behaviour in which perception is directly linked to action. At this unattended level, the world is directly perceived through a perception layer (the thalamus) which routes sensory information into episodic memory. Episodic memory encodes previously encountered situations, narrative structures and well-learned plans or behaviour schemas. It expresses the relative salience of all episodic associations, mapping these into both procedural and working memory. Procedural memory, which includes central and motor pattern generators, is heavily interconnected with episodic memory and encodes procedural motor skills.

There is strong neuroanatomical evidence to support the existence of, and distinction between episodic and procedural memory. Both memory subsystems appear to be hierarchically organised [Fuster, 1995; Baddely et al, 2002; Tulving, 1993]. Episodic memory has its hierarchy directed from perceptual links at the bottom of the hierarchy, to its connections with working memory at the top. Likewise procedural memory has its hierarchy directed from the connections with working memory down to its connections with motors via the thalamus. In addition both episodic and procedural memory are composed of several hierarchies [Fuster, 2003], episodic memory has a separate hierarchy for each sensory mode (i.e. touch, smell etc.), whilst procedural memory has a separate hierarchy for major effector systems (i.e. arms, legs etc.).

In the Attentional Architecture the hierarchical representation of the world and the reactive response to it is achieved by distributed collections of nodes (clusters of neurons) within episodic and working memory, respectively. For example the recognition of a letter within a word and a word within a phrase may involve the activation several ‘recogniser nodes’ in episodic memory (these nodes are based on the ‘grapheme’ and ‘sememe’ units used by Plaut & Shallice [1993] to simulate deep dyslexia; see *Chapter 5* for a more detailed description).

The activation pattern in episodic memory produces a corresponding pattern of activity in ‘recogniser nodes’ in procedural memory. Continuing the example of word recognition, in the context of reading allowed (that is, the current intention is to pronounce what one sees) this pattern encodes for the motor actions needed to pronounce the sequence of phonemes for a given word or phrase. In this account of the relation between episodic and procedural memory, and in the context of a current intention (see working memory, below) the semantics of the world are represented by the connections between collections of recogniser nodes distributed across memory subsystems. Almassy & Sporns [2001] have

shown that learning to perform simple categorisation of perception is semi-automatic and does not require sustained executive attention.

The precise nature of the linkage between episodic and procedural memory is unclear. Baerends [1976] and Tyrell [1993] both argued that behaviour (and therefore memory) were organised hierarchically into a single structure indicating the possibility that the links between episodic and procedural memory may be physically structured in some way. Fuster [2003] argues that there are, indeed, extensive recurrent links between each level of the two hierarchies. Deco and Rolls [2005] found that forward links in associative (episodic) and motor (procedural) memory were biased in favour of the corresponding backward links.

In the architecture proposed here, there are several distinct types of connections between episodic and procedural memory that provide the model with distinct features:

- Connections from episodic to procedural memory serve to prime procedural recogniser nodes (actions) based on salient episodic information [Fuster, 2003]. As actions have to be disinhibited by the contention scheduler, there is little chance that this priming will initiate movement, however these connections are able to bias motor (and action) selection where there is little or no competition from higher levels in procedural memory
- Pathways, from episodic to procedural memory allow the co-ordination of complex movements; these connections are routed through the thalamus and protected from contradictory episodic associations by the contention scheduling (the basal ganglia)
- Connections from procedural memory to episodic memory serve to prime episodic memory on the basis of salient action schemas [Frith, 1998]. This provides the architecture with a prediction or ‘forward model’ of anticipated sensations and perceptions that episodic memory can use to detect deviations or errors in actions as they happen or as they are simulated by procedural memory rehearsing movements (i.e. the final motor outputs remain inhibited)

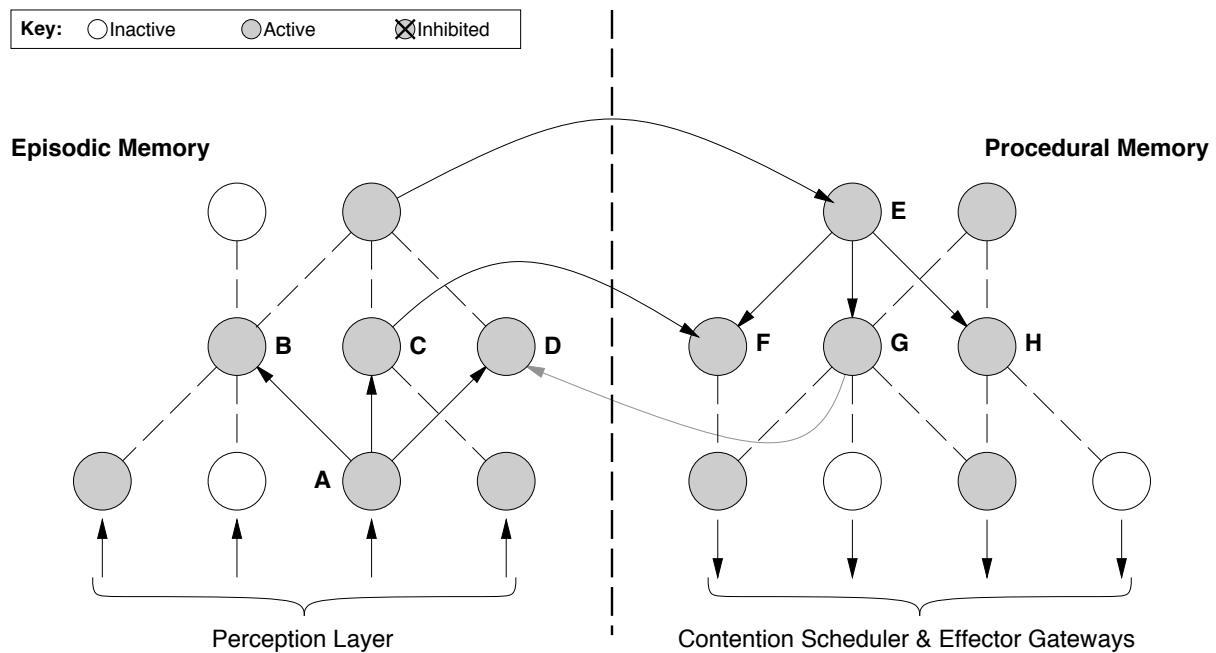


figure 4.3 The structure of the reactive links of episodic and procedural memory and their interconnections. The diagram shows normal divergent memory activation across episodic and procedural memory. In episodic memory node A, stimulates nodes B, C and D, whereas Node E in procedural memory attempts to activate nodes F, G and H. In addition nodes E and F in procedural memory are connected to pathways (i.e. node F has a pathway connection from node C and an excitation connection from node E) directly from nodes in episodic memory. Node D in episodic memory is primed by the activity of node G

The essential functional and architectural features of episodic and procedural memory in the Attentional Architecture are illustrated in figure 4.3. Perception induces a pattern of excitation in episodic memory. Low-level nodes represent ‘atomic’ associations or episodes and higher-level nodes ‘fuse’ these, thus representing richer and more abstract biographical associations with perceptual input. This pattern of excitation, in turn, excites nodes at corresponding levels in procedural memory. High-level nodes represent compound actions. Procedural memory propagates activation to low-level nodes representing ‘atomic’ actions. At each level in episodic and procedural memory nodes representing contradictory reactive associations compete for expression through the reactive systems contention scheduling mechanism (vertical dotted line in the diagram). Procedural memory links back to episodic memory serve to prime episodes that are related to highly salient actions. This ‘forward modelling’ cues episodic memory to respond to action-related changes in the perceived world. For purposes of clarity, mutually inhibitory links between layered nodes in episodic and procedural memory respectively are not shown here. (Note: the ‘nodes’ in this and later illustrations should not be interpreted as individual neurons, nor the links between them as individual axons. Rather, such nodes are usually functional clusters of neurons that serve to ‘recognise’ an input pattern and propagate a response pattern at some level of salience.)

4.3. Working Memory and the Supervisory Attentional System

4.3.1. Working Memory

Working memory, as defined originally by Baddeley & Hitch [1974], is a store of short term information. Neuroanatomical evidence for working memory is unclear and currently has several definitions, Shah & Miyake [1999] give an account of the seemingly contradictory evidence for its existence, function and structure. Baddeley & Logie [1999] continue to argue for the working memory being a short term information system separate from, but adjacent to, episodic memory. Cowan [1999] proposes that working memory is actually long-term memory in a heightened activation state.

Like both episodic and procedural memory, working memory is composed of large numbers of 'recogniser' nodes. Anatomically, working memory appears to reflect an architectural convergence of the hierarchies of episodic and procedural memory [Fuster, 1995]. (Indeed, Fuster takes a very particular view in proposing that that all memory and executive functions are actually organised into (and part of) the two hierarchical primary subsystems, episodic and procedural memory. Fuster [1995] further argues that each level of the hierarchy has recurrent connections to the other and that connections from both episodic and procedural memory converge in pre-frontal cortex).

For the purposes of the architecture to be developed here, these different perspectives are each complied with in some measure without actually adopting one particular model. Working memory provides the Supervisory Attentional System (SAS) with representations of current goals and intentions, encodings of highly salient episodes from episodic memory, and induced salient actions from procedural memory. The connections from both episodic and procedural memory represented into working memory are governed by contention scheduling so there is selection of competing salient episodes and related actions and once selected for, they tend to persist. This convergence into working memory is governed by contention scheduling, thus, all highly salient non-contradictory episodes and possible actions encoded within working memory are 'observable', in parallel, by the SAS at any one time. This architectural configuration provides an implementation of Broadbents [1958], 'selective filter theory' which proposed that attention and the identification of physical stimuli in higher order cognitive processes are physically limited [Lachter et al, 2004].

Though the detailed hierarchical structure of working memory isn't known, the physical structure of episodic and procedural memory, together with the requirement for access to working memory by multiple parallel systems of nodes, allows speculation that it is layered into hierarchical 'planes', each plane interconnected with several above and below it. The output of working memory is connected directly to procedural memory, in a similar manner to the connections between episodic and working memory, and is known as the 'preparatory set' [Fuster, 2003]. *Figure 4.4* illustrates the functional and architectural relationship between working, episodic and procedural memory. Interaction between episodic memory and procedural memory through working memory is complementary to the automatic activation shown in *figure 4.3*.

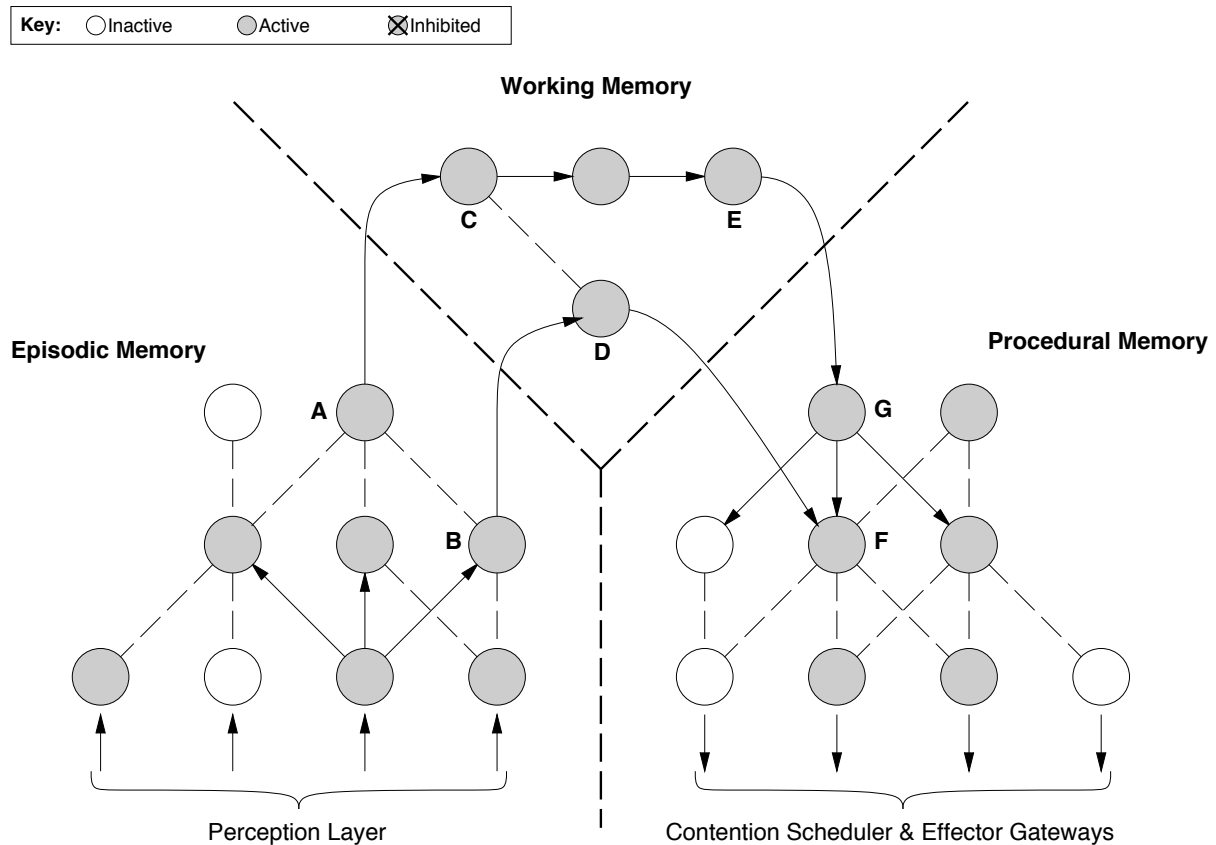


figure 4.4 An illustration of how episodic, working and procedural memories interact. Nodes A and B in episodic memory are connected to corresponding layers in working memory (nodes C and D respectively). Nodes D and E (the preparatory set) are also connected to corresponding layers in procedural memory (nodes F and G)

Figure 4.5 expands upon figure 4.4 to show more clearly how the top layers of episodic memory relate to layers in working memory via contention scheduling. The response of working memory reflects input from episodic memory, procedural memory (via episodic memory) and internal representations of current goals and intentions (developed by higher cognitive systems, including the SAS Generator and maintained by working memory). Working memory has direct access to high level procedural memory nodes. The resulting activity in procedural memory is a reflection of both the reactive pathway of figure 4.3 and of the pattern of activation in working memory in figure 4.4 and figure 4.5. Having episodic and procedural memory connected through working memory may provide the attentional system with a means to combine episodes and representations of motor functions that can be 'ranged through' or even 'played forward', in the absence of behavioural expression, thus providing a means to predict, or look ahead [Clark and Grush, 1999; Shanahan, 2005].

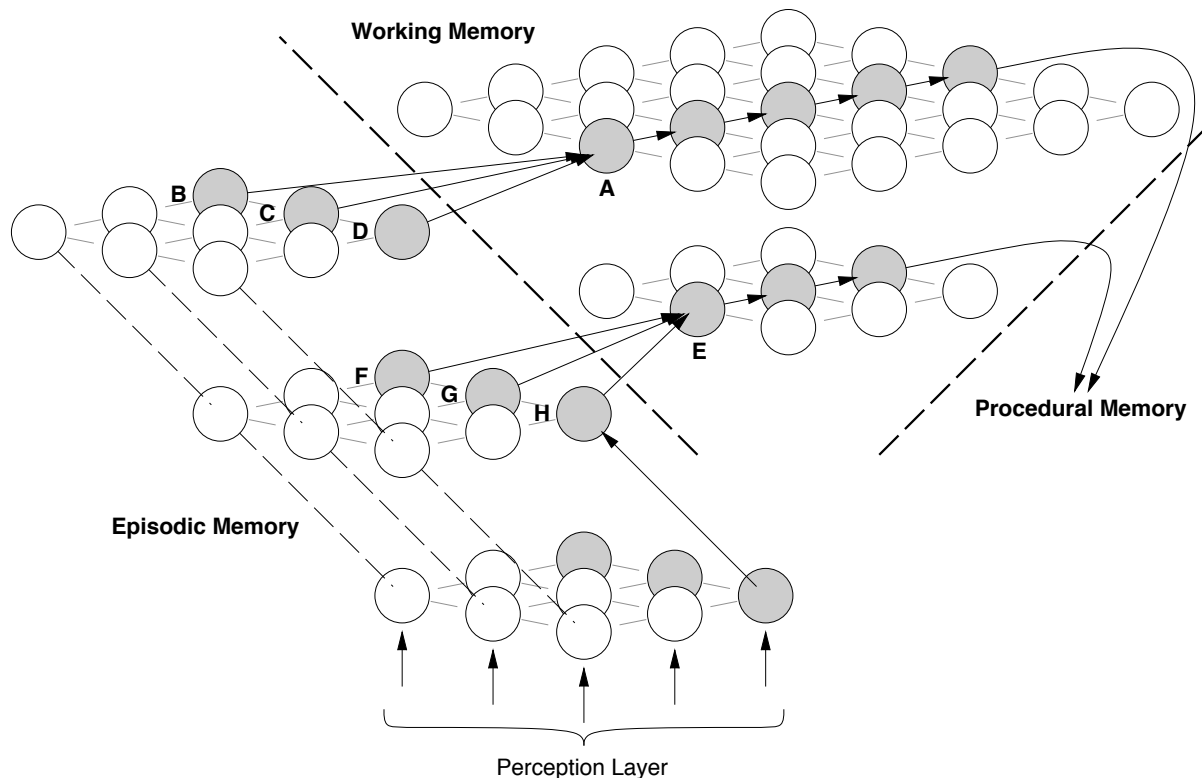


figure 4.5 Expanded diagram showing the speculated hierarchically structured episodic memory interacts with the presumed layered structure of working memory. Nodes **B**, **C** and **D** compete for access to node **A** via the contention schedulers (not shown). In a lower layer access to node **E** is being competed for by nodes **F**, **G** and **H** respectively. The output of working memory is connected to the hierarchical structure of procedural memory in a similar manner

4.3.2. *The Supervisory Attentional System (SAS) and Sub-functions: Monitor, Modulate, Generate*

Neuroanatomical evidence for the SAS/Executive is very limited indeed although very high-level functions, such as planning, are almost invariably associated with the frontal and prefrontal cortex, the association is rarely more specific in structural terms. It may turn out to be the case that some or all of its sub-functions are distributed, as suggested by Fuster [2003].

Functionally, the SAS has three subsystems, corresponding to the three sub-functions most recently specified by Shallice & Burgess [1998]. The SAS Monitor can be thought of as a system for detecting unfamiliar or novel circumstances, defined as a ‘departure from expectation’ and especially a departure of intended from expressed action. By monitoring working memory (and its inputs and outputs), the SAS Monitor is able to observe the unfolding pattern that derives from observation of the perceived world (from episodic memory), currently intended actions or behavioural goals, expressed actions (from procedural memory and effector contention scheduler), and the outcome of expressed action perceived through the changed state of the world (made apparent in a subsequent state of episodic memory). If the SAS monitor detects departure from expectation, it generates an arousal or ‘warning’ signal to the SAS Modulator. This would appear to be the point at which deliberative attentional effort is invoked (a moment of ‘bringing to consciousness’).

The SAS Modulator expresses willed intention by attenuating the salience of inappropriate (unplanned) actions and potentiating the salience of intended tasks. Note that this limited mechanism (modulation) deals only with minor levels of variation of action selection; it only allows expression of actions that are related to current intentions and are relatively salient. If the intervention of the SAS modulator proves inadequate to the novelty of a situation, the SAS Generator must be able to produce novel approaches to problem solving. Strategies might include creating entire novel sequences of intended actions (new plans).

The SAS Generate sub-function is associated with the production of innovative plans to deal with novel situations. In the Attentional Architecture presented here, this third sub-function is not implemented. Its output is represented by an encoding in working memory of an externally determined goal requiring a sequence of actions.

4.3.3. Working Memory and the SAS

The operation of the SAS and working memory are intimately related (indeed, Fuster [2003] would make no distinction). For the purposes of the Attentional Architecture described here, a distinction *is* made. *Figure 4.6* illustrates the interactions between episodic memory, working memory, the SAS functions Monitor and Modulate and the contention scheduler when a highly salient episodic node, which is not related to the current goal held in working memory, becomes expressed. Node **A** (representing the reactively intended behaviour) has primed node **C** (as in *figure 4.3*). Both node **A** and node **C** have gained expression in working memory (nodes **E** and **D**, respectively) via contention scheduling (dotted line) and are thus subject to monitoring by the SAS monitor. However if the activity of node **B** is significantly increased (e.g., through a strongly triggered habitual response) and gains expression via the contention scheduler in working memory, the SAS Monitor (node **F**) senses the difference between intended and activated behaviour (a different pattern of activity in nodes **D** and **E**). An arousal or warning signal is generated which induces the SAS Modulator (node **G**) to attenuate the ‘error’ behaviour. It does so potentiating both node **C**’s excitatory inputs and node **B**’s inhibitory inputs and via the contention scheduler (nodes **H** and **I**, respectively), increasing the likelihood that node **B** is inhibited relative to node **C**. Nodes **H** and **I** being adjacent in the contention scheduler architecture have mutually inhibitory links so that excitation in one node suppresses the other.

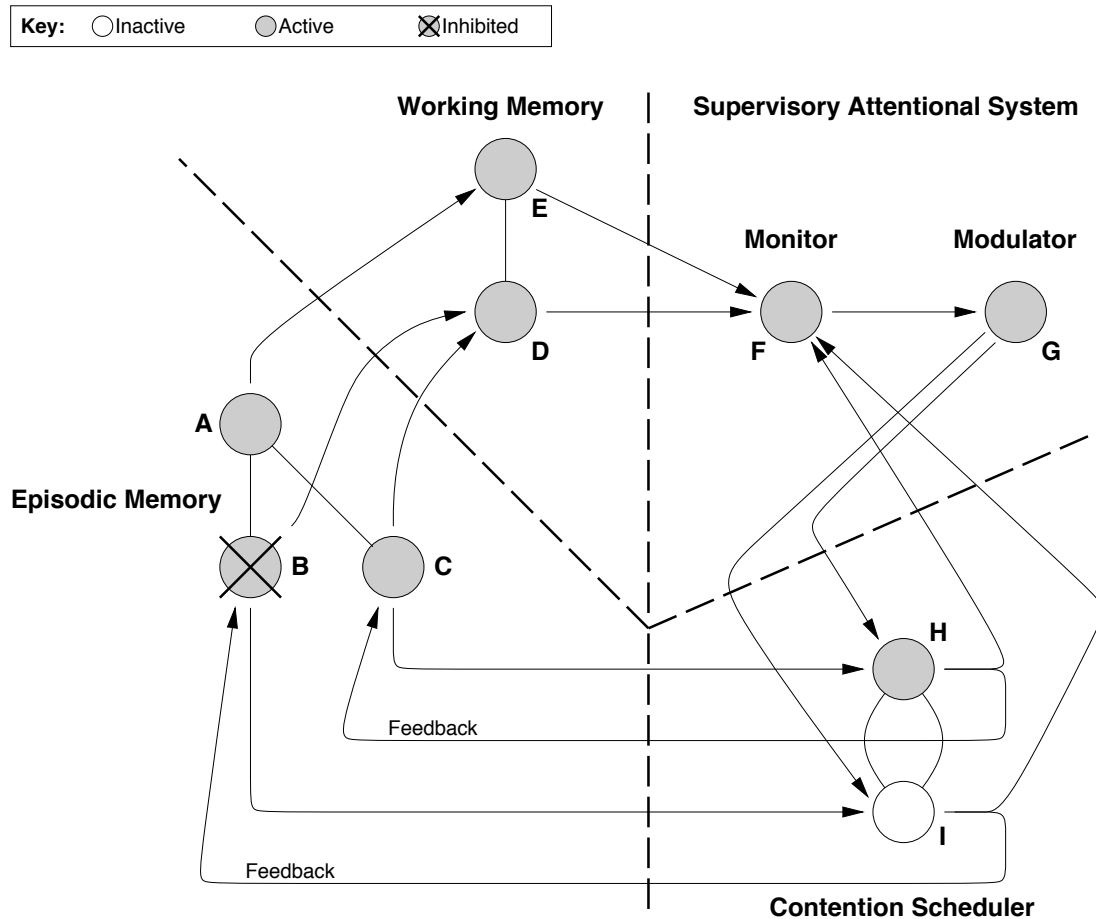


figure 4.6 Interactions between episodic, working memory and the Supervisory Attentional System

4.4. The Affective System and Learning

How behaviour producing structures and processes might be modified on the basis of experience to produce improved performance has long been an open question in agent design [Maes, 1994]. A considerable body of work on reinforcement learning has examined the role of extrinsic signals and of internal signals based upon ‘drives’ (e.g. hunger). In respect of intrinsic signals, Shallice & Burgess [1998] have proposed that the Supervisory Attentional System plays a role in generating temporary behavioural ‘schemata’ which are then somehow learnt by behaviour generating systems if they perform in a positive way. A hypothesis explored in the Attentional Architecture presented here is that the level of supervisory attentional activity is a candidate for an intrinsic learning signal which reinforces learning of reactive behaviour which initially requires attentional resources.

The affective system, which includes the limbic system and, specifically, the hippocampus [Fuster, 1995, pp. 36-40] is intimately associated with the sensation of emotion. Activity in the limbic system is expressed in a number of ways. It indirectly induces release of hormones which trigger changes (e.g., in heart rate, respiratory rate, distribution of blood flow, etc.) associated with emotional sensations. More importantly in this context, it is also associated with the release of neurotransmitters and neuromodulatory chemicals such as noradrenalin, 5-hydroxytryptamine, dopamine and

acetylcholine. In contrast to neurotransmitters, neuromodulators are slow acting, diffuse, and non-specific in information content [Cohen, Dunbar & McClelland, 1990]. Neuromodulatory chemicals released by the limbic system, and especially the hippocampus, are known to play a role in promoting episodic learning.

Anatomically, the hippocampus is known to take input from episodic memory, procedural memory and the pre-frontal cortex (which is here inferred to represent elements of the SAS) and provides connections back from the hippocampus back into episodic memory and by implication, working memory. The hippocampus has a direct influence on the creation of long-term (episodic) memory [Shapiro & Eichenbaum, 1997]. Lesions to the hippocampus can significantly impair the subsequent creation of autobiographical memories, though retrieval of memories established prior to the damage remains largely unaffected.

A simple model of the affective system has been included in the model to investigate the hypothesised relation between supervisory attentional activity and learning. In the model considered here, the connections between the SAS and the hippocampus allow for episodic and working memory learning by using the level of modulatory attentional activity in the SAS as the basis for a hippocampal signal that mediates reinforcement of the association between the currently willed action and the current active pattern of episodic and procedural salience which is, itself, a reflection of SAS modulatory activity. Over time, the intrinsic reinforcement of this association means that a diminishing level of attentional effort will be needed to re-establish its likelihood of expression in response to the given perceptual cues.

Episodic memory and procedural memory appear to have different learning mechanisms. The limbic system and especially the hippocampus are known to be involved in episodic learning and probably working memory [Burgess et al., 1995; Shastri, 2002]. Processes involved in procedural learning are less certain, but may integrate attentional effort and self-priming positive feedback derived from selection for expression by the contention scheduler.

Within the Attentional Architecture the simulated limbic system (hippocampus) is used principally as a mechanism that ensures that a neuromodulatory substance is released when the SAS Modulator is active (see *figure 4.7*; this figure extends the scenario as shown in *figure 4.6*). As the SAS works to attenuate node **B** and potentiate node **C**, it also induces release of a diffuse neuromodulatory chemical signal from the limbic system. The presence of this substance reinforces the inhibitory connection between node **C** and node **B** in episodic memory.

(The role of the hippocampus in memory and behaviour extends beyond the account implemented here. It is known that the hippocampus plays a significant role in the acquisition, integration and retrieval of memory, specifically in episodic and working memory, in association with other parts of the limbic system [Fuster, 1995; Aggleton & Pearce, 2001; Shapiro & Eichenbaum, 1997].)

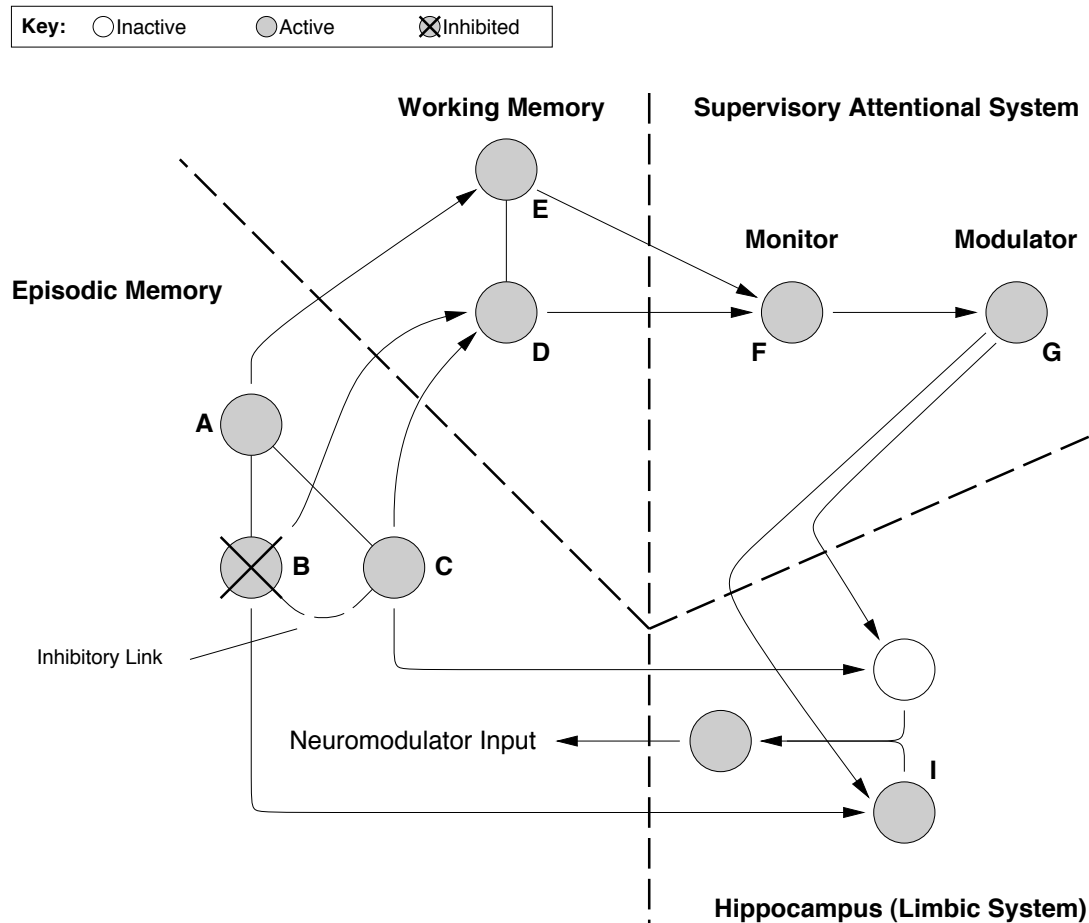


figure 4.7 Learning in episodic memory

In contrast to episodic and working memory learning, the learning of new procedural skills does not seem to depend directly upon the hippocampus, thus a second mechanism is needed to support this form of learning. We have already seen that the SAS modulates the expression of behaviour by potentiating and/or attenuating the salience of episodic nodes contending for expression by the contention scheduler. The output of working memory (the preparatory set) is connected directly to procedural memory. However, this in itself does not appear to constitute a learning signal. Once the contention scheduler finds and expresses an activated procedural pattern, reinforcing (thalamo-cortical) feedback to procedural memory seeks to promote the persistence of the expression of this activity. This self-priming mechanism serves to avoid constant switching of nodes arising from very minor change in the perceived environment [Snaith and Holland, 1991]. This feedback mechanism is a plausible basis for a reinforcement signal to procedural memory that results in the learning of an attended response.

4.5. High-level Summary Account of the Attentional Architecture

In this final section the functional-architectural account of the integrated system is summarised in a series of figures: *figure 4.8*, *figure 4.9* and *figure 4.10*. These annotated figures attempt to convey the scale of complexity and integration of the system. They represent a point of reference in respect of the subsequent chapters.

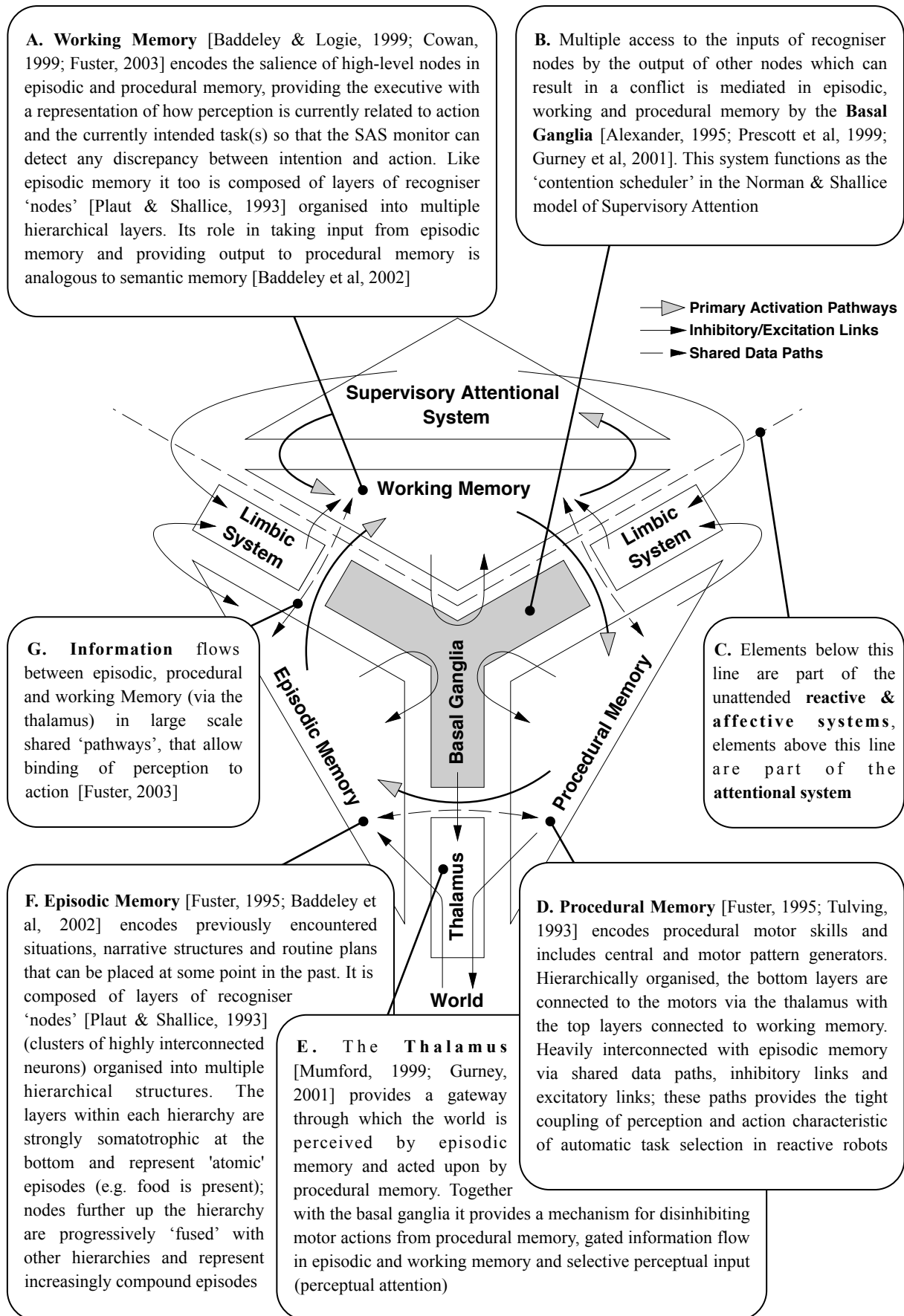
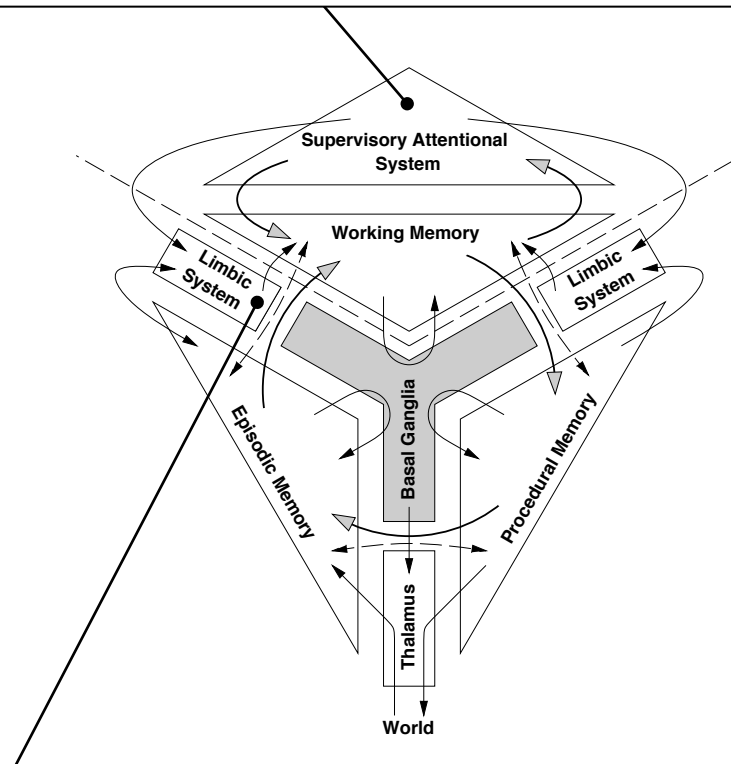


figure 4.8 The Attentional Architecture showing the primary components and connection paths

H. The [Norman & Shallice, 1986; Shallice & Burgess, 1998] **Supervisory Attentional System**, is composed of three primary subsystems:

- The **SAS Monitor** can be thought of as a system for detecting novel circumstances by monitoring working memory. The SAS Monitor nodes receive input from three sources: the salience levels of the perceptually driven associations from episodic memory, that express the reactive intention of the system; the salience of behaviours which are allowed expression by the basal ganglia (contention scheduler), i.e. the reactive response of the system; and the salience of intended behaviours from working memory. The SAS Monitor raises an arousal (warning) stimulus to the SAS Modulator if a behaviour is expressed which has not been planned. Once triggered, the arousal signal continues to be generated until the conflict is resolved by changes in the relative salience levels of the input behaviours
- **SAS Modulation** clusters generate output patterns that modulate the signals from memory clusters into the basal ganglia (contention scheduler) so that intended behaviour is potentiated and other behaviours (competing for access) are attenuated. It is important to recognise that this does not guarantee the selection of the intended behaviour, as this risks overriding behaviours strongly and appropriately triggered by the environment, e.g., those designed to prevent harm to the machine or its surroundings
- If the intervention of the SAS Modulator proves inadequate to the novelty of a situation, the **SAS Generator** must be able to produce novel approaches to problem solving. Strategies might include creating entire novel sequences of intended actions



I. The **Limbic system** (including the **Hippocampus**) is intimately associated with the sensation of emotion, indirectly inducing release of hormones which trigger physiological changes (e.g., in heart rate, respiratory rate, distribution of blood flow, etc.) associated with emotional sensations. The overall level of activity of the SAS modulator provides the basis for a signal to the limbic system. Low levels of activity in the SAS Modulator are associated with routine (minimally-attended) responses to familiar circumstances; high levels of activity reflect attentional effort in the face of unfamiliar circumstances. When the limbic system becomes active due to raised attentional activity, a modulatory chemical signal is propagated through the entire neural network, reinforcing links from active nodes in episodic memory that are not suppressed by the SAS to active nodes which are suppressed by the SAS. Over time, this results in a reinforcement of the association between the attended to episode and expressed action so that, in future, reduced, or zero attentional effort will be required for the appropriate action to occur automatically

figure 4.9 The emotional and supervisory components of the Attentional Architecture

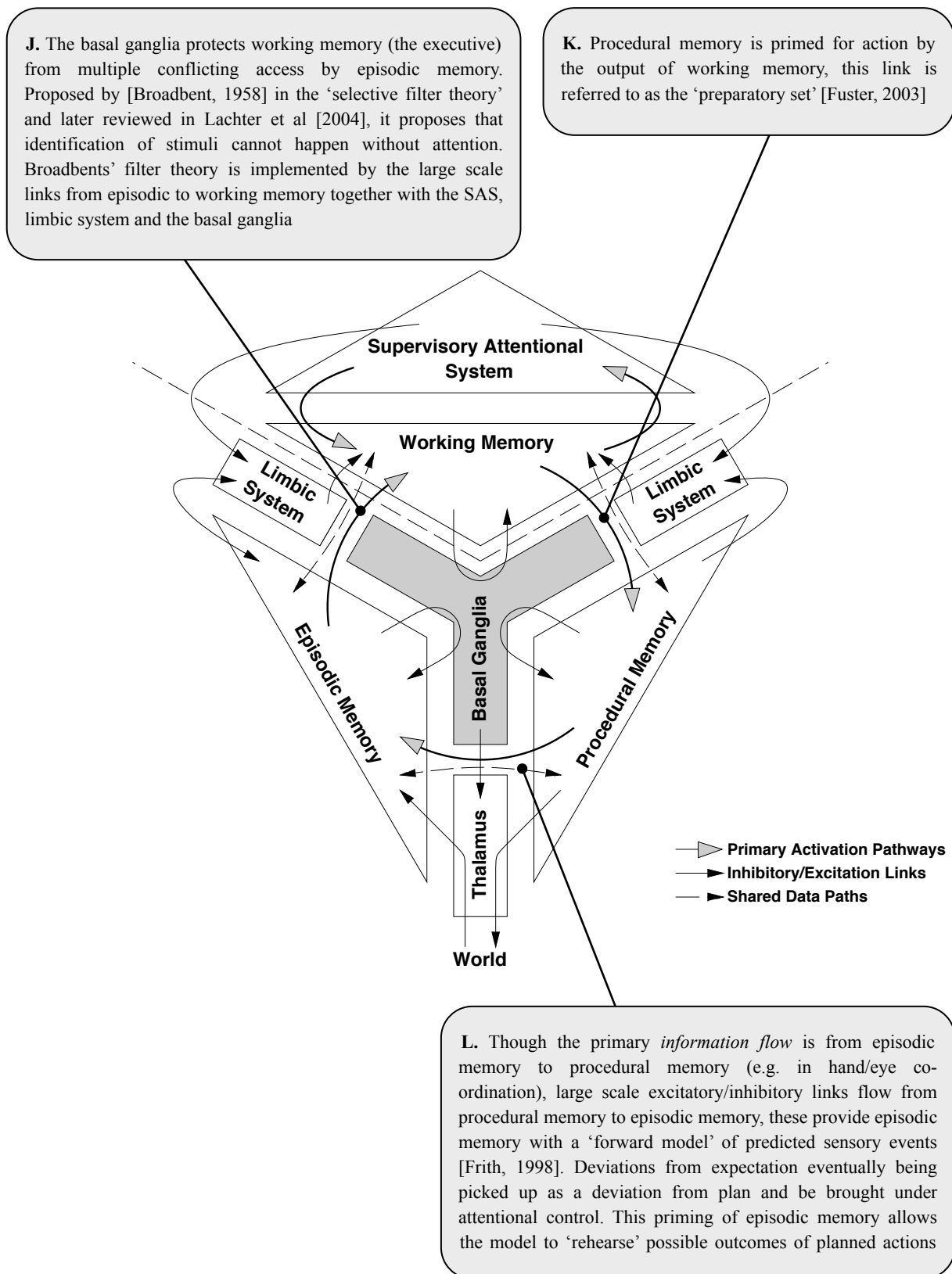


figure 4.10 Interaction points between episodic, working and procedural memory

Chapter 5

5. Detailed Design

This chapter revisits the high-level account given in *Chapter 4*, in order to arrive at a position from which it is possible to proceed towards implementation of the system (see *Chapter 6*). The approach adopted is to identify a basic functional design unit from which much of the system can be constructed.

The structure of the subsystems and their interconnections will echo an apparent principle underpinning neural systems in the brain according to which there are large numbers of locally and densely interconnected neurons (clusters) which are less densely connected to yet other clusters, both local and distant. Analogues of such clusters will contribute to the basic functional building blocks of the architecture. The first section of the chapter describes in greater detail the ‘recogniser nodes’ used within episodic, procedural and working memory. Following this, the second section describes the design of working memory and the third section describes the design of the two Supervisory Attentional System functions featured: SAS Monitor and SAS Modulate.

5.1. Elements of Detailed Design

In many neural models the neuron itself provides the minimal design unit. This is feasible if the system to be developed has a relatively small number of neurons; it is desirable if neuroanatomical evidence is available to inform such modelling; and it is considered essential if the modelling seeks to explore the functional properties of a system at the level of the architectural relationships between individual neurons. In the work presented here, the number of neurons is large, the neuroanatomical evidence is limited to large-scale structures and much is inferred, and the functional properties of the system being explored arise from the architectural relationships between structures implementing major sub-functions. Accordingly, the minimal design unit adopted here is one that encapsulates a manageable level of functional performance but at a low enough level for the properties of the system to emerge rather than be entirely predetermined.

In the account given in *Chapter 4* the elements providing the major sub-functions (episodic, procedural and working memory) have been described chiefly in terms of nodes, and specifically ‘recogniser nodes’. Recogniser nodes are based on units developed by Hinton et al when developing an architecture which exhibited different forms of dyslexia [Hinton, Plaut and Shallice, 1993; Plaut and Shallice, 1993]. In that work, grapheme and sememe units, which recognised word morphology and expressed word semantics, respectively, were considered as densely connected recurrent clusters of neurons which processed an input pattern to produce an output pattern of some level of salience.

There is no corresponding evidence that recogniser nodes actually exist within the brain, though some uniformity of structure has been found in the neocortex; consisting of six primary layers [Fuster, 2003, p. 62], layer IV of the neocortex receives input from the thalamus, layers II and III are connected to other areas of the neocortex and layer V and VI to various other structures including the thalamus and basal ganglia; these layers are in turn composed into units or ‘parcels’ of highly interconnected columns of cells separated by areas with low cell density and few connections.

Recogniser nodes superficially resemble the components of Fuster’s [2003, pp. 14-16] ‘Cognits’, these components, also called ‘cognits’¹ encode information in its component nodes (neural networks) and the relations (connections) between them. The topology of large scale cognit ‘collections’ encodes cognitive operations and retrievable memory; nodes can be part of several (various) cognits, with learning facilitated by the formation of new cognits and modification of old ones. Unlike Fuster’s cognits, recogniser nodes within the Attentional Architecture are based primarily on pattern matching [Hinton, Plaut and Shallice, 1993; Plaut and Shallice, 1993]. They are organised to reflect the hierarchical structural relations of Tyrell [1999] and the gated and contention scheduled information flows of Prescott [1998].

¹ The definition is recursive, cognits may be composed of sub-cognits.

Behaviours are coded in the current pattern of activity in all active and suppressed nodes and the connections between them within the entire model in a similar fashion to general cognition in Fuster's cognitive architecture.

Whilst recogniser nodes feature in episodic, procedural and working memory, no such basic unit is evident in respect of the SAS. Accordingly, the neural structures of the SAS are developed separately in order to provide the functionality required in respect of the memory subsystems: monitoring and modulation.

In the remainder of the section, a detailed account of the design of recogniser nodes is given, followed by an account of the SAS.

5.2. *Recogniser Nodes and Pattern Matching*

From a design perspective, it is possible to specify a hybrid 'recogniser node', encompassing the main design attributes derived from Plaut & Shallice [1993], Tyrell [1993] and Fuster [2003], which can be used as a basic building block for *all* of the memory subsystems (episodic, working and procedural) in the model.

It will become clear that recogniser nodes are functional entities that map onto (across) a number of related anatomical structures, including those of memory and the basal ganglia:

- *Salience based hierarchies* [Tyrell, 1993] as described for episodic, working and procedural memory [Fuster, 2003]
- Large scale, parallel, *information pathways* between nodes and memory subsystems, governed by contention scheduling

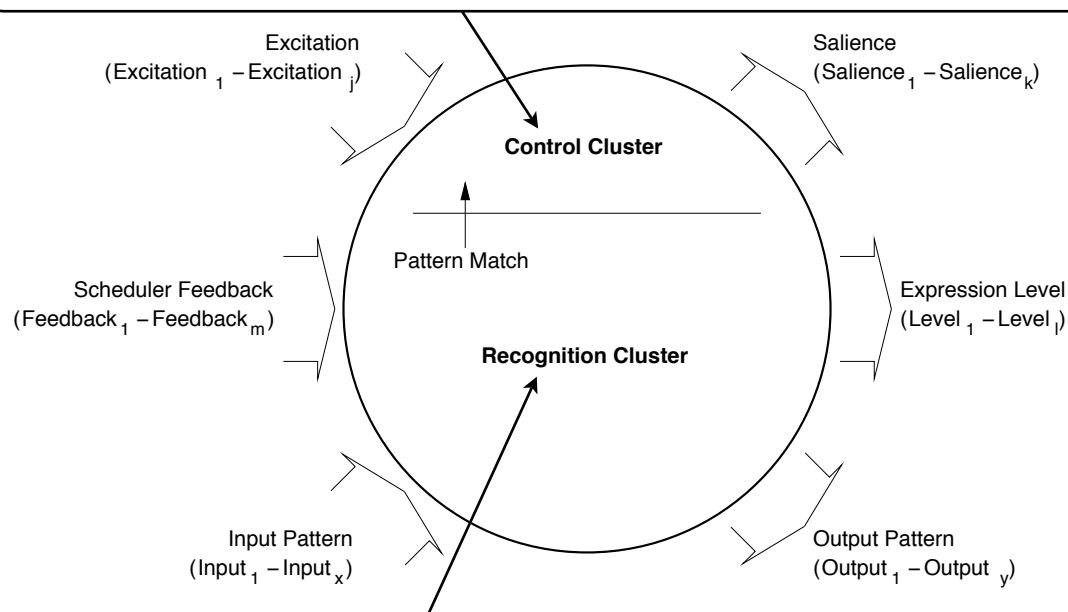
A generic recogniser node is illustrated in *figure 5.1*. Information (pattern) processing is achieved through a recognition cluster:

- An **Input Pattern**, consisting of parallel input signals ($I..x$) generates an **Output Pattern** consisting of ($O..y$) parallel output signals connected to information pathways, which, if shared, are subject to contention scheduling
- A signal internal to the node, **Pattern Match**, indicates the strength of association (similarity) between the output pattern arising from the current input and the nearest output pattern that constitutes one of the *categories* learnt by the node.

The functioning of the hierarchical structure within which the node sits is achieved through the use of 'control' signals (interpreted and generated using a control cluster for each node), the definitions of which relate to each other:

- **Excitation** is the weighted sum of excitatory/inhibitory salience signals from $(1..j)$ other recogniser nodes in the system. (All links between nodes have a weight of 1.0 unless modified as a result of attention based learning, see *section 4.4*.)
- A nodes own **Salience** (communicated to $(1..k)$ other nodes) is a function of the strength of pattern matching in the node (**Pattern Match**), the level of **Excitation** received from other recogniser nodes and the level of reinforcing feedback when expressed through any contention scheduling mechanism (**Scheduler Feedback**).

The control cluster is a separate but connected set of neurons to the recognition cluster. It is used by each node to interpret salience signals from the memory hierarchy (**Excitation**) and the contention scheduler (**Scheduler Feedback**) and to generate two primary output signals: **Expression Level** (to the contention scheduler) and **Salience** back into the memory hierarchy and the SAS.



Recognition clusters take input patterns and generate output patterns. An internal '**Pattern Match**' signal from the recognition cluster is an indicator to the control cluster as to how well an input pattern matches current known output patterns (categories).

figure 5.1 A recogniser node, used as a basic building block for episodic, working and procedural memory subsystems within the Attentional Architecture

Competition between nodes and *dynamic* information flow are supported by two further control signals and the linking of a node's input and output signals to information pathways (in the thalamus) where they compete for expression:

- For nodes that compete for output access to (shared) information pathways an **Expression Level** output ($1-l$) is connected to a 'winner takes all' contention scheduler. If a node requires access to several information pathways, this signal may be connected to several contention schedulers (The difference between **Expression Level** and **Salience** is that the latter incorporates **Scheduler Feedback**).

- Nodes competing for expression through contention scheduling receive feedback and the input signal **Scheduler Feedback** sums the input ($I-m$) from all the contention schedulers that the recogniser node is connected to. This is the mechanism that supports the persistence of output once selected for by the contention scheduler

To control the flow of information through a recogniser network, the input and output signals **Excitation**, **Salience**, **Expression Level**, **Scheduler Feedback** and the internal signal **Pattern Match** can be grouped into a simple control cluster for each node.

Figure 5.2 illustrates how the signals to and from a recogniser node relate to contention scheduling and expression of output (e.g. to effectors or to working memory) via ‘thalamic’ gateways. In the figure recogniser nodes **B** and **C** are connected to a shared pathway and compete for expression at the ‘output neurons’ beyond the thalamic gates. The contention scheduler governing the shared pathway selects one of the nodes competing for expression based upon the relative strengths of the ‘**Expression Level**’ signals. The ‘**Scheduler Feedback**’ connection from the contention scheduler to the selected node also provides the dis-inhibition signal to the thalamic gates, which, in this illustration results in expression of the output pattern of node **C**.

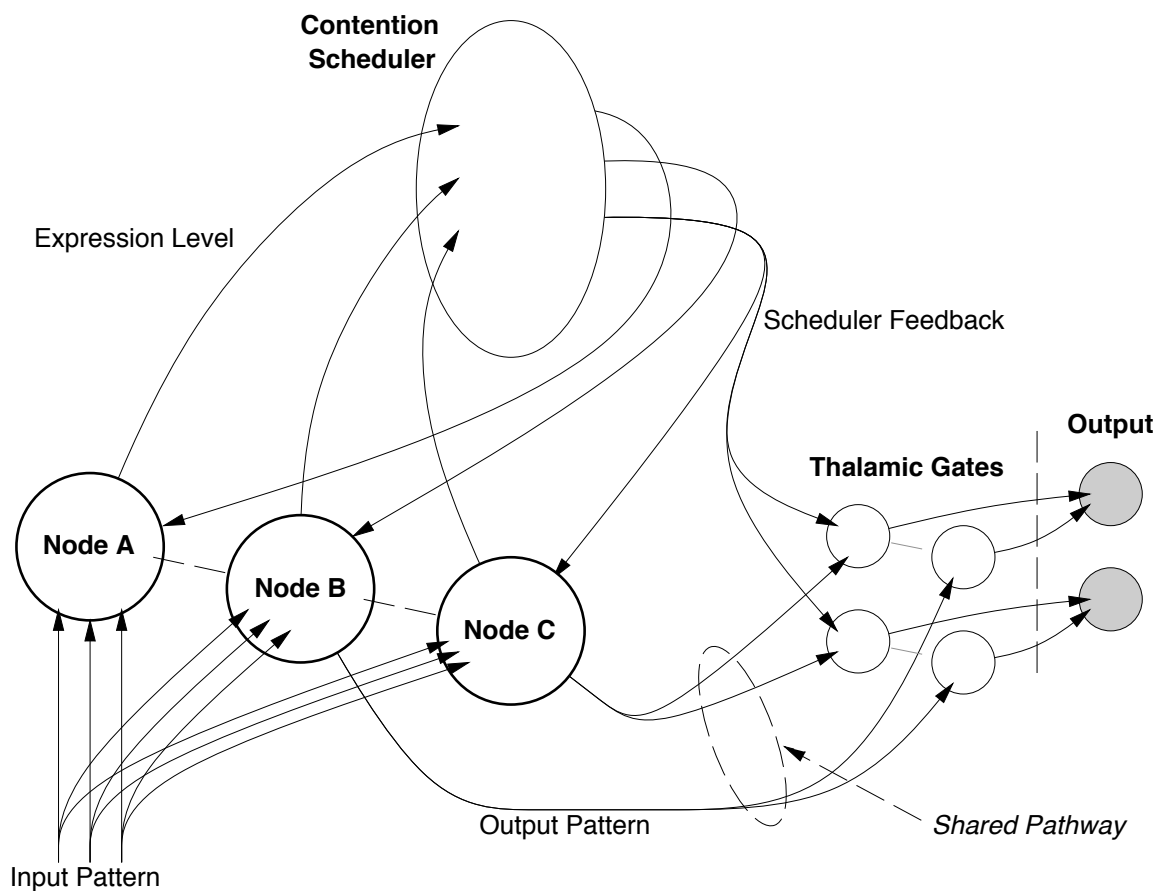


figure 5.2 An illustration of how recogniser nodes interact with the contention scheduler and thalamic gates.

Figure 5.3 illustrates how recogniser nodes combine in episodic memory to produce hierarchical associative processing. As can be seen from the fragment of episodic memory represented, sensors are connected (via the thalamus) to the bottom layer (layer 1) producing input patterns to node A and node B). Nodes A and B process the input pattern to give an output pattern. This output pattern, together with a signal relating to the strength of this input-output association (**Pattern Match**) in the node is propagated to nodes in the next layer in links labelled **Saliency** (e.g. from node A to nodes C and D). Higher nodes sum the **Saliency** signals (e.g. node C from nodes A and B) from lower nodes to compute the incoming **Excitation**. At highest levels of episodic memory where there is a convergence towards working memory [Fuster, 2003] Nodes C and D compete for access to node E through the contention scheduler and thalamus (not shown in this diagram, refer to figure 5.2)

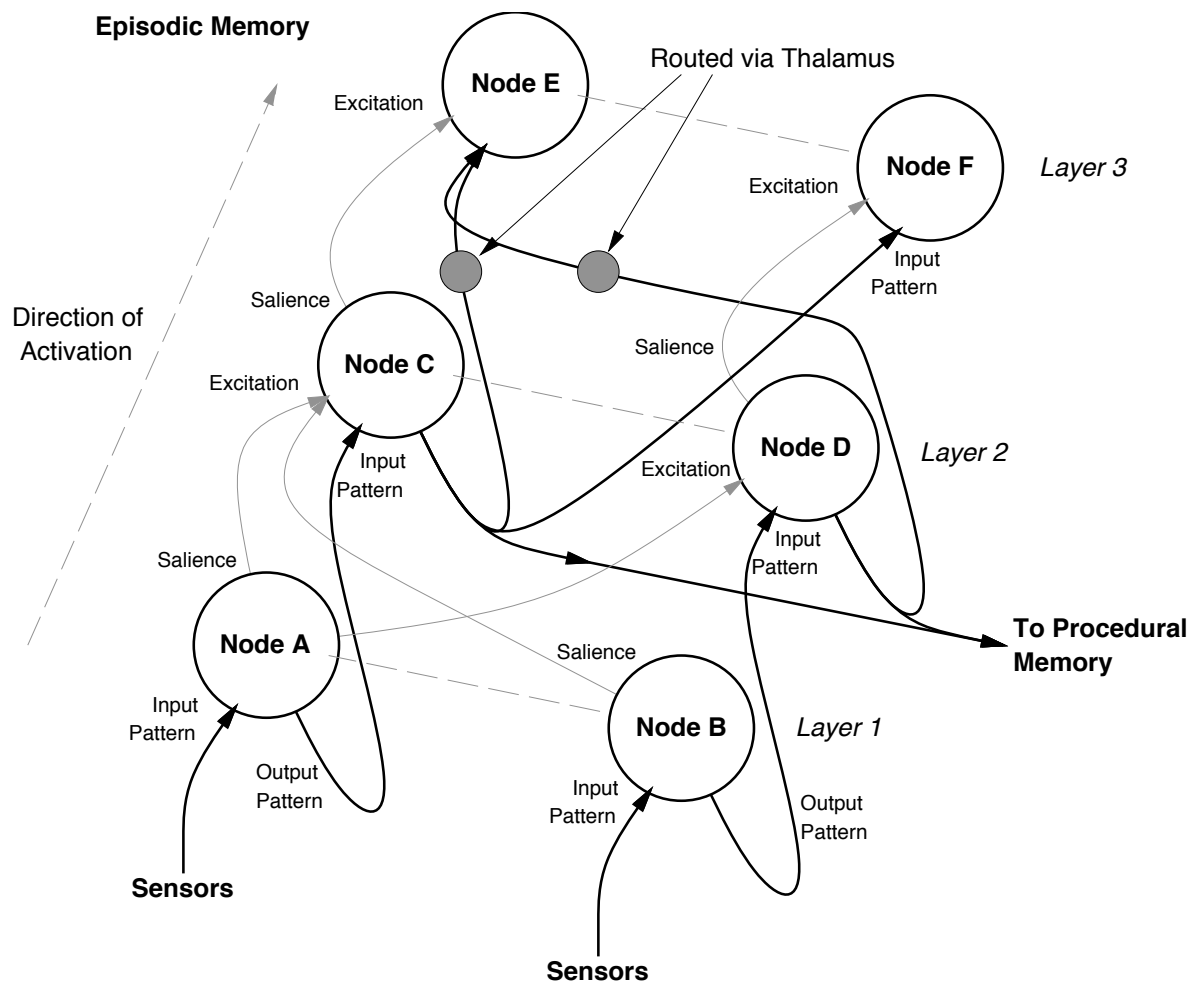


figure 5.3 A diagram of the proposed structure of episodic memory composed of recogniser nodes.

Procedural memory has a very similar structure to episodic memory (as shown in figure 5.3) except its activation structure is from higher layers (receiving input from working memory) to lower layers with the lowest layers connected to effectors via the thalamus. In terms of the layered connections between episodic and procedural memory, episodic memory primarily propagates output patterns computed by the nodes but procedural memory propagates expression signals thus implementing the priming of episodic nodes associated with highly salient behaviours.

5.2.1. A Taxonomy of Recogniser Nodes

From a design perspective it is possible to consider the roles of recogniser nodes in the network and develop a simple taxonomy of such nodes, specified with reference to how the input pattern, output pattern and control signals are internally interpreted (A good description of how a number of convergent and divergent neural networks can be used to categorise information is given by Amassay and Sporns [2001, p. 129]). The taxonomy of recogniser nodes adopted for design purposes is:

- **Stream Processors:** At the lowest levels of episodic and procedural memory, these recogniser nodes take a continuous data stream and generate an associated output. Because of their position and function they do not compete for output and computationally they can ignore **Excitation**, **Expression Level**, **Scheduler Feedback** and the internal signal **Pattern Match**. The **Salience** output can be ignored by nodes that are immediately above stream processors (it will always be “high”)
- **Bottom Up Recognisers:** These nodes become increasingly salient as the **Pattern Match** internal input indicates that the **Output Pattern** generated by the **Input Pattern** is well matched to one or more output patterns (categories) that the node has previously learnt. If the **Output Pattern** is connected to a shared information pathway the node will have to compete for expression. *Figure 5.4* details the activity pattern
- **Top Down Recognisers:** Effectively ignoring **Pattern Match** until the **Excitation** input is also active, these nodes become increasingly active when other parts of the memory hierarchy excite (recruit) the node and the **Output Pattern** is well matched to one of more output patterns that the node has previously learnt. Like bottom up recogniser nodes these networks have to compete for activation if the **Output Pattern** is connected to a shared pathway. One use of this type of node is to *search* memory, e.g. using attention to enhance or suppress initial high level categorisation matches. Its activity pattern is shown in *figure 5.5*
- **Pattern Generators:** Connecting the **Output Pattern** of a node to the **Excitation** input of a number of lower level top down recogniser nodes produces a *sequence* or *pattern of excitation* in the hierarchy. Inputs to the pattern generator can be used to *prime* or *trigger* the next pattern of a recurring cycle.
- **Observers:** If the **Salience** outputs of several nodes are connected to the **Input Pattern** of a node, the current state of the network hierarchy that the node is connected to can be observed, captured as a state, matched to expected patterns of activity and propagated as an output pattern. If the input is a combination of desired (from pattern generators) and actual perceived activation states these nodes could be used as a general ‘satisficing’ mechanisms [Simon, 1957; Gigerenzer et al., 1999].

5.2.2. Bottom Up Recogniser Nodes

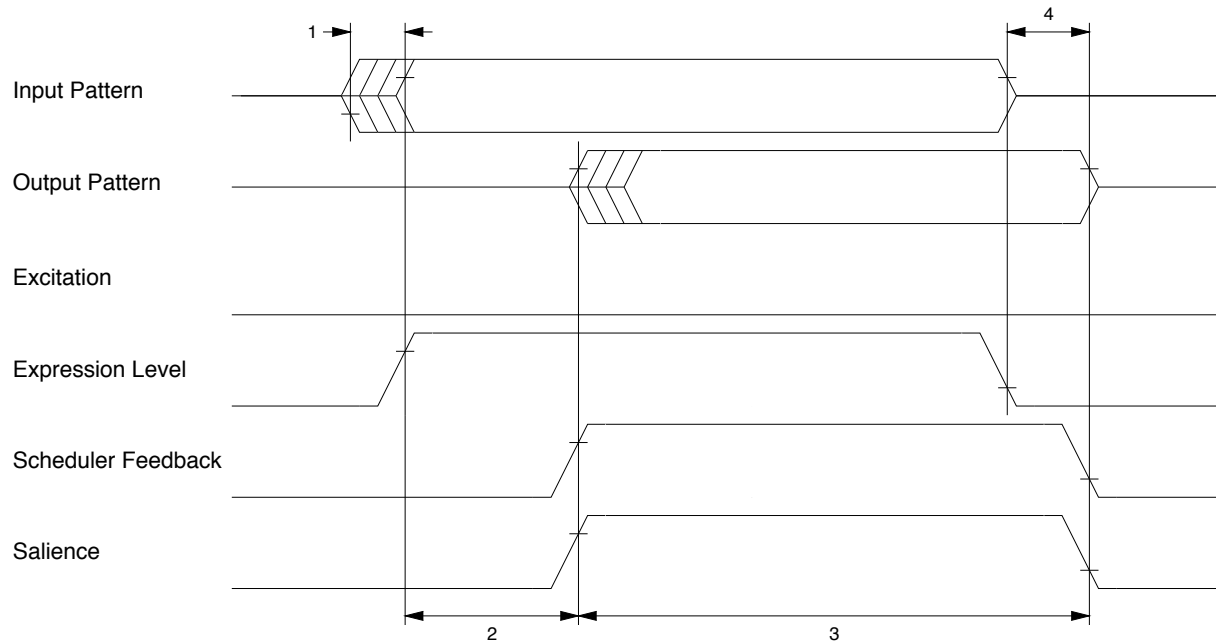


figure 5.4 The activity pattern of a 'bottom up' recogniser node, recognising an input pattern and becoming salient (active). The sequence of events is as follows:

- 1) The Input connections (1 through to x) becomes active over a given period
- 2) At some point the input pattern is recognised (by the recogniser network) and the **Pattern Match** input into the control network is raised (not shown); in a 'bottom up' node this immediately causes the **Expression Level** output to become active (shown by its rising edge)
- 3) A time period after **Expression Level** is raised, the node is granted activation by the contention scheduler shown as the rising edge of **Scheduler Feedback** and its outputs are gated onto the data-paths that it is connected to (shown by the rising edges of the **Output Pattern** connections). At the same time the node indicates its current level of salience as the rising edge of the **Saliency** output
- 4) Once the **Input Pattern** connections become inactive (or the match of the input pattern fails), **Expression Level** also becomes inactive. Persistence of activation caused by the contention scheduler can cause a delay between the falling edge of **Expression Level** and the subsequent falling edges of **Scheduler Feedback** and **Saliency**

5.2.3. Top Down Recogniser Nodes

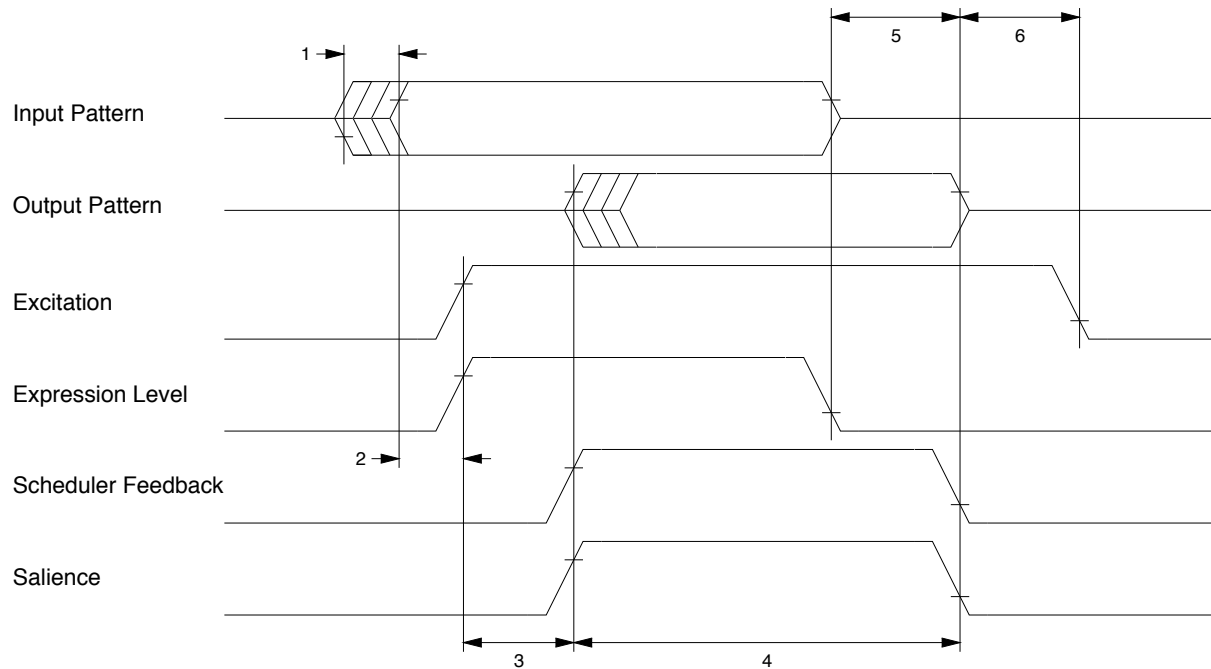


figure 5.5 The activity pattern of a 'top down' recogniser node, being requested to become salient after recognising an input pattern. The sequence of events is as follows:

- 1) The **Input Pattern** connections (1 through to x) becomes active over a given period
- 2) At some point the input is recognised (by the recogniser network) and the **Pattern Match** input into the control network is raised (not shown); however in a 'top down' node this does not immediately cause the **Expression Level** output to become active. Instead the control network is biased by the **Excitation** input such that it will only raise the **Expression Level** signal when it is active
- 3) A time period after **Excitation** and **Expression Level** have been raised, the node is granted activation by the contention scheduler shown as the rising edge of **Scheduler Feedback** and its outputs are gated onto the data-paths that it is connected to (shown by the rising edges of the **Output Pattern** connections). At the same time the node indicates its current salience as the rising edge of the **Salience** output
- 4) Once the **Input Pattern** connections become inactive (or the match of the input pattern fails), **Expression Level** also becomes inactive
- 5) Persistence of activation caused by the contention scheduler can cause a delay between the falling edge of **Expression Level** and the subsequent falling edges of **Scheduler Feedback** and **Salience**
- 6) At some point later the **Excitation** input is lowered (shown by its falling edge), after this point input patterns are again ignored

5.3. Working Memory

Figure 5.6 illustrates the arrangement of recogniser nodes in working memory. It is an elaboration of the working memory part of figure 4.4 and figure 4.5 from Chapter 4. Working memory is probably layered, the layers reflecting the results of contention scheduling on the upper layers of episodic memory. In this figure the input patterns to nodes **C** and **D** are the outputs from the most salient nodes in the highest level of episodic memory and the input patterns to nodes **A** and **B** are from the most salient nodes in the next highest level of episodic memory (see figure 4.4 and figure 4.5). Thus, working memory has a representation of the currently most salient episodic association. Activation in the layers of working memory flows from connections with episodic memory towards procedural memory, defining the preparatory set in nodes **E** and **F**, respectively (these nodes also resemble the ‘afferent copy’ of motor commands as proposed by Frith [1998, p. 186]). This arrangement is a reflection of Fuster’s [2004] assertion that the highest level of procedural memory *is* working memory and it meets the requirement that the working memory provides the SAS with a representation of the currently most salient episodic associations and the reactively associated responses (behaviours).

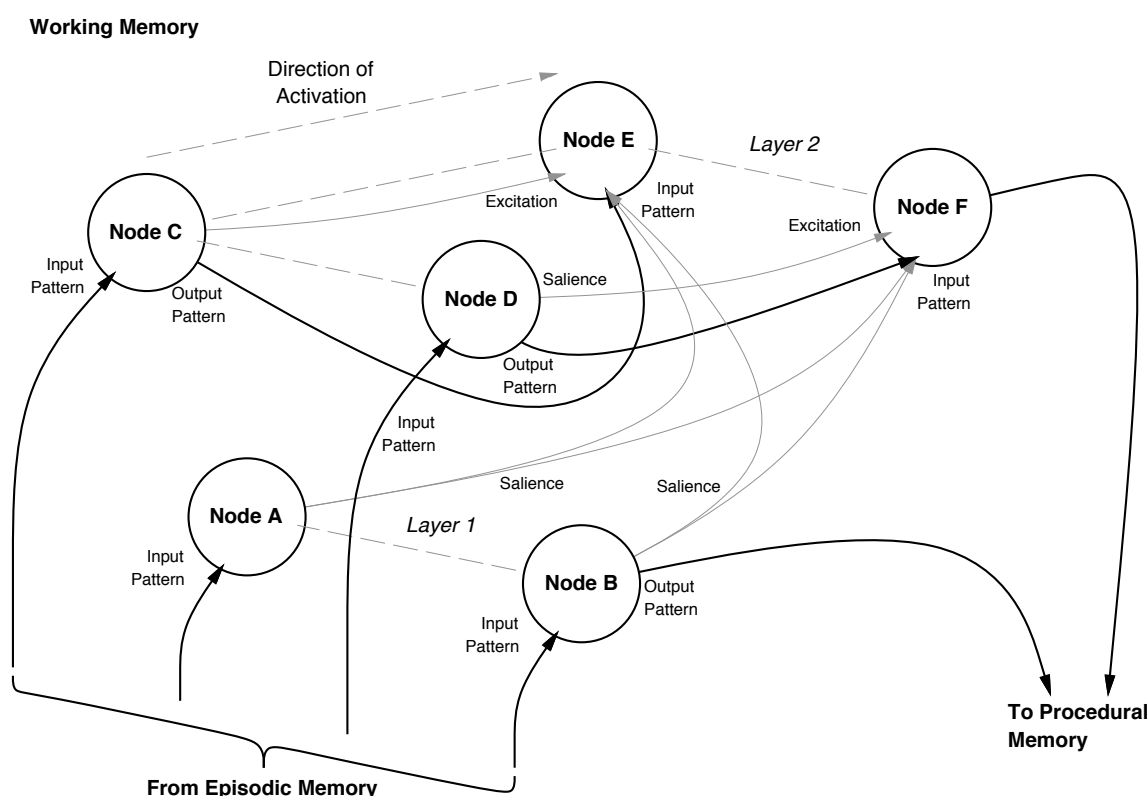


figure 5.6 The speculated architecture of recogniser nodes in working memory, interacting with other recogniser nodes in episodic and procedural memory. Activation in these layers is directed from connections with episodic memory to procedural memory (the preparatory set). Node **A** and node **B** comprise layer 1, with Nodes **C**, **D**, **E** and **F** comprising layer 2. Nodes **A** and **B** can analyse the activation pattern of episodic memory they are connected to and attempt to excite procedural memory based on this input. Patterns within episodic memory will excite Nodes **A**, **B**, **C** and **D**. Thus, nodes in higher levels of working memory, such as nodes **E** and **F**, integrate information from more than one layer of episodic memory

5.3.1. Memory Composition and Recogniser Types

Outlined in figure 5.7 is the speculated dominant distribution of recogniser node types in episodic, procedural, and working memory

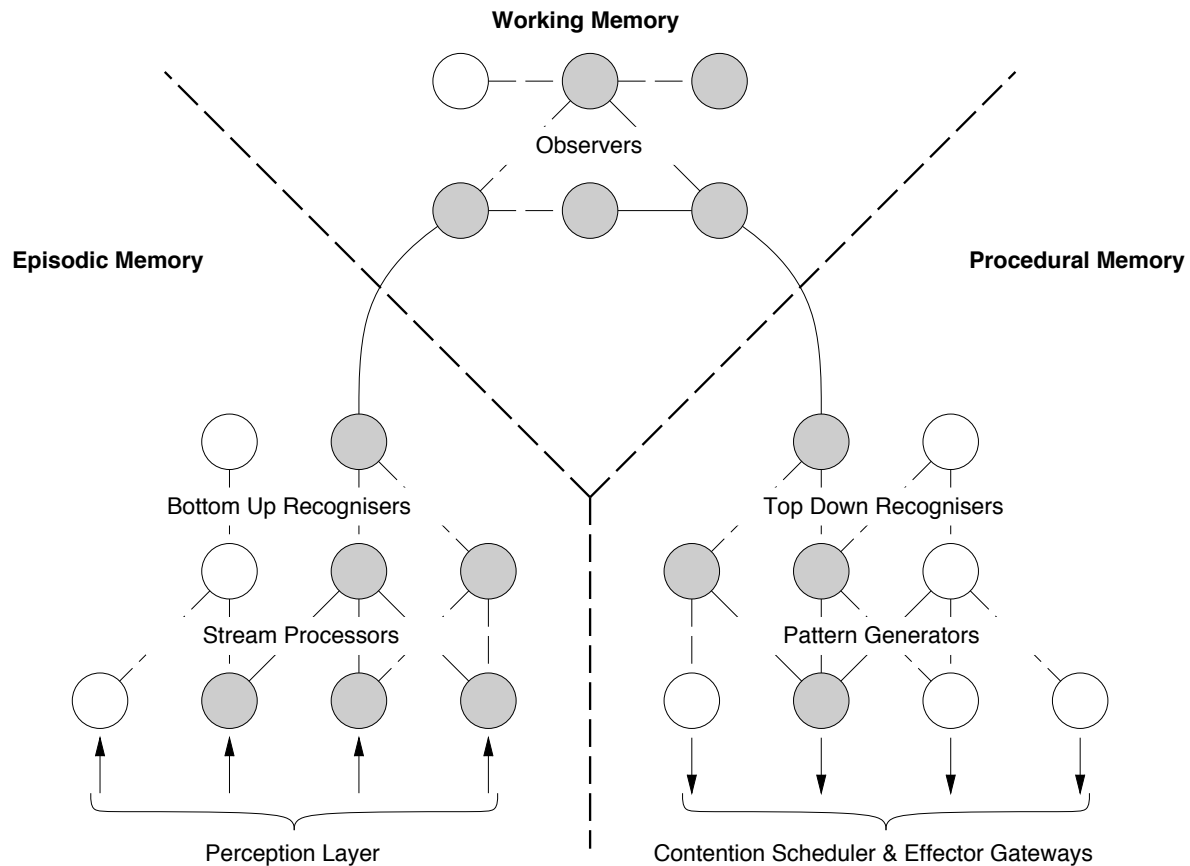


figure 5.7 Recogniser types and their assumed primary location in each memory subsystem

5.4. The Structure of the Supervisory Attentional System

There is little or no neuroanatomical evidence in relation to the SAS and certainly nothing is known about its internal structures. For the purposes of the model the SAS has been kept as a separate element of the architecture. However, it may be the case, as Fuster [2004] suggests, that executive functions (and therefore the SAS) may be directly integrated into the memory subsystems.

As observed behaviour is a combination of episodic association, intended behaviour (in working memory) and expressed actions (from procedural memory) it might appear that the SAS could modulate *all* of memory; this is thought unlikely as a large scale neuroanatomical structure connected to the cortex would have already been observed.

Functionally, the SAS can be defined in relation to the memory subsystems that it is necessarily connected to [Frith, 1998]:

- The SAS Monitor has to detect a difference between anticipated reactive activity (from episodic memory) and expressed activity (output from contention scheduling of behaviours in the basal ganglia, and possibly of the preparatory set in procedural memory) and, when a sufficient difference is encountered, it has to raise an arousal or warning signal to the SAS Modulator
- The SAS Modulator has to produce a modulatory pattern of potentiation and/or attenuation across the pattern of reactively anticipated behaviour in a way that increases its correspondence to intended/willed behaviour
- Learning is induced during SAS Modulator activity to try to ensure that the willed behaviour becomes the automatically expressed behaviour over time and in the model proposed here that is achieved through excitation of the affective system (especially the hippocampus)
- Each recogniser node that has connections to the contention scheduler (and hence can be inhibited or excited by the SAS Modulator) and has inputs from other nodes such that its activation can be ‘willed’ can be monitored by the SAS modulator
- Each SAS Monitor is paired with a SAS Modulator
- The SAS Generator (not implemented in this work) would be influenced by SAS Modulator activity in a way that created varied or entirely novel patterns of willed behaviour

Using the input and output signals designed for the memory subsystems, the corresponding connections needed to support the operation of the SAS are given in *figure 5.8*; its target activity pattern is shown in *figure 5.9*:

- The summed **Excitation** signal of a recogniser node is connected to the SAS Monitor’s **Excitation** input, such that the SAS Monitor can see the anticipated reactive activity of the node
- The SAS Monitor’s **Scheduler Feedback** input is connected to the contention scheduler (s) that the recogniser node uses, so that the SAS Monitor can see the expressed activity of the node
- When the **Scheduler Feedback** differs sufficiently from **Excitation** an arousal or warning signal is generated that triggers the SAS Modulator
- The **Expression Level** output of a recogniser node is connected to the SAS Modulator’s **Expression Level** input, so that the SAS Modulator can see the level of expression in each node attempting to gain expression via contention scheduling
- The SAS Modulator’s **Modulation Output**, once sufficiently excited by an arousal or warning signal from the SAS Monitor, attempts to inhibit the expression of the recogniser node (at the contention scheduler) until the level at which the node seeks expression subsides (by observing **Expression Level**)

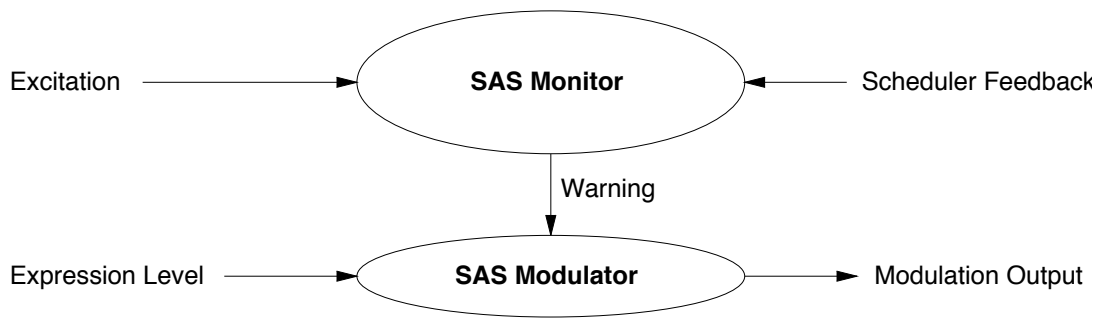


figure 5.8 An overview of a SAS Monitor and SAS Modulator

Activity in the SAS Modulator is proportional to the extent to which it is invoked by activity in the SAS Monitor. When invoked, modulation of the input to contention scheduling continues until the conflict inducing the response from the SAS Monitor is resolved and the warning signal subsides. This can be brought about by changes in the relative stimulus levels of the nodes in and associated behaviours in working memory, which can arise in a number of ways, but principally through a change in the perceived state of the world

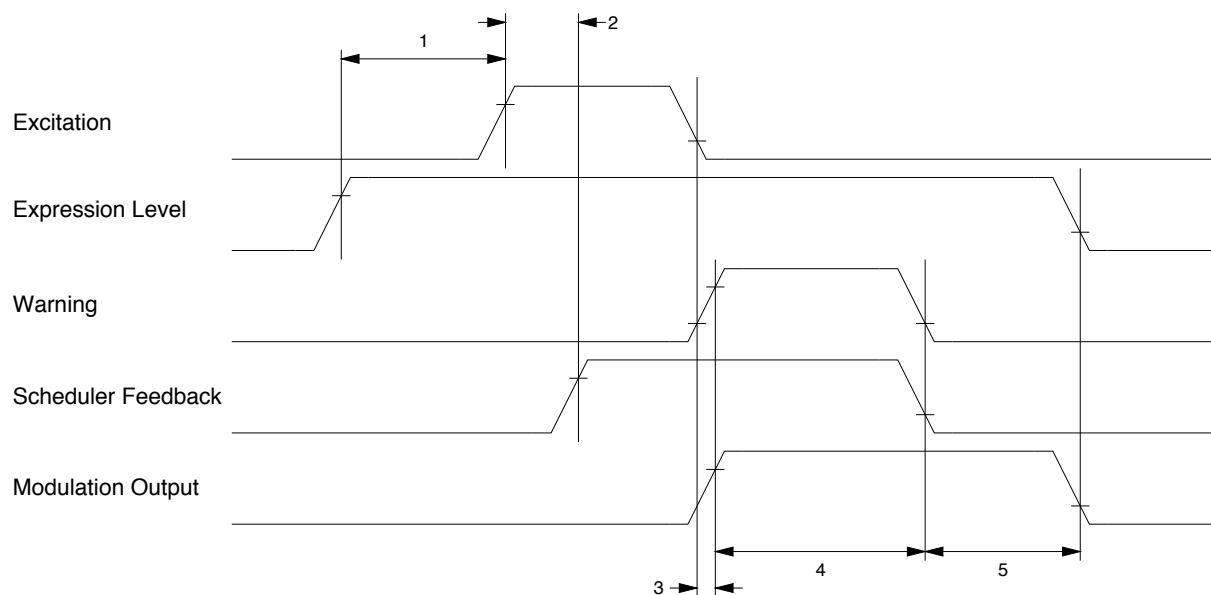


figure 5.9 The activity pattern of an individual SAS Monitor and Modulator node. The rising edge of **Expression Level** (1) indicates that the recogniser node that the SAS node is monitoring has become active (i.e. it has recognised a familiar pattern on its inputs and is requesting expression), though it is currently unplanned, the SAS ignores this input until the rising edge of **Excitation** (2) together with at some point its corresponding scheduler feedback (the rising edge of **Scheduler Feedback**). After actual expression the recogniser node has completed its task within the behaviour and the **Excitation** falls (3). As the activity of the recogniser node is now no longer planned, the SAS Monitor activates the **Warning** signal (3) causing the SAS Modulator to activate (4). Due to the modulation of the contention scheduler, the induced learning caused by the modulation or the environment, the recogniser node becomes deselected by the contention scheduler and loses expression (5) at the same time as the **Warning** signal becomes inactive; however as the recogniser node is still active, the **SAS Modulation Output** continues its activity until the falling edge of **Expression Level** is detected

SAS Monitors are designed to detect a difference between planned behaviour and actual behaviour, signalling any departures (arising from novelty or unfamiliarity) to the SAS modulator; this then modulates these behaviours in an attempt to correct the error. Behaviours in the Attentional Architecture are mapped to collections of recogniser nodes, but the SAS Monitor has little reason to monitor nodes which cannot be ‘willed’ (either the **Excitation** input is ignored by the node or the **Salience** output of a node is fixed) or monitor nodes that it cannot modulate, i.e. the node does not share a pathway with any other node and is therefore not connected to the contention scheduler (the **Expression Level** output or **Scheduler Feedback** input are ignored). Hence the domain of the SAS (the regions of memory over which it acts) can be speculated about as shown in *figure 5.10*, reflecting Lurias [1970] assertion that a complex function such as attention is spread across several brain areas.

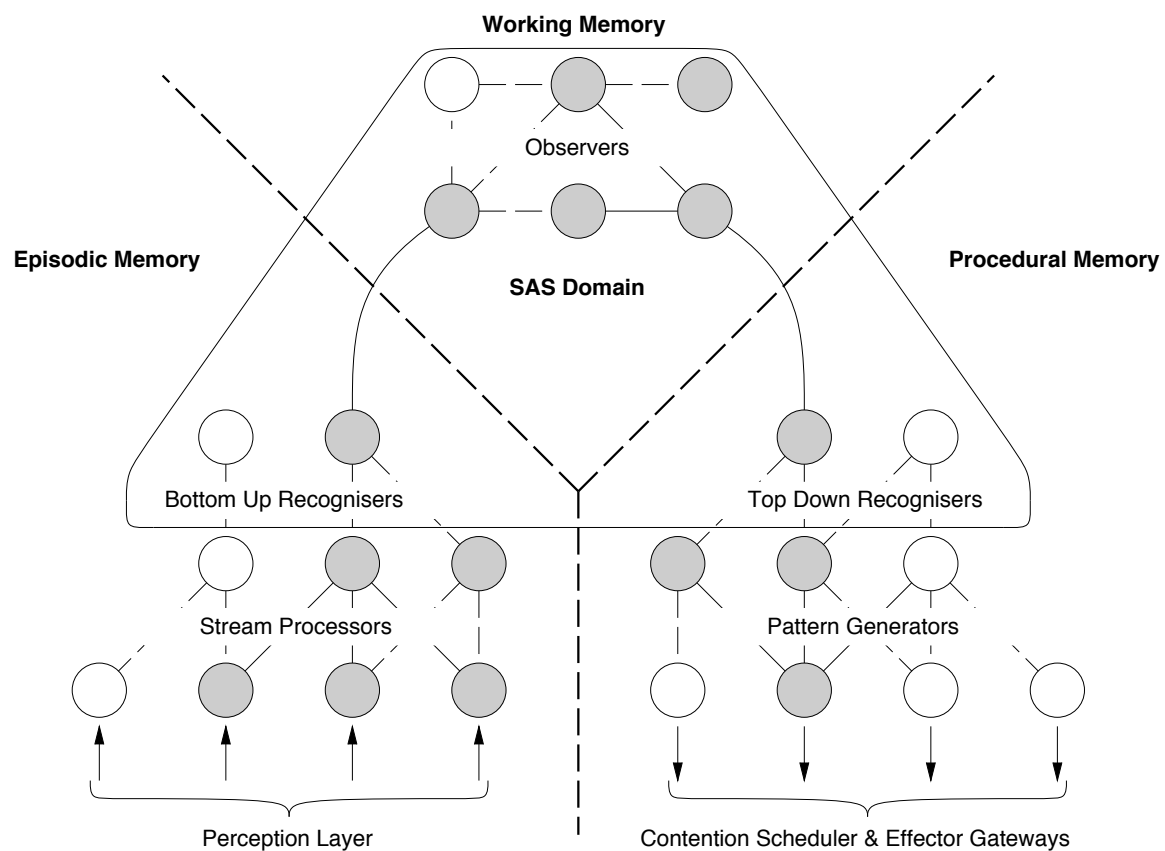


figure 5.10 As the SAS cannot monitor nodes that cannot be planned for or modulate nodes that do not use a contention scheduler, a modified version of *figure 5.7* can be created, showing the memory areas potentially connected to and controlled by the Supervisory Attentional System

Reflecting this in the current model, modulation presented to the contention scheduler from the SAS attenuates activation of a recogniser node. Whilst the SAS Modulator is involved in enhancing the likelihood of expression of ‘willed’ behaviours not currently expressed, it is thought this may only be operative in conjunction with the SAS Generator (not currently part of the model) as otherwise modulation is likely to result in simply enforcing stereotypical behaviour.

5.5. Summary

Figure 5.11 illustrates the relationship of the modelled elements of the SAS to the other elements of the control architecture. In fact, the arrangement illustrated in figure 5.11 suggests the basis for a single functional ‘building block’ or ‘Generic Behavioural Mechanism’ which can be scaled up to provide an implementation of the architecture. Such a building block encapsulates episodic association, working memory, SAS monitoring and modulation, procedural response, contention scheduling and thalamic gating.

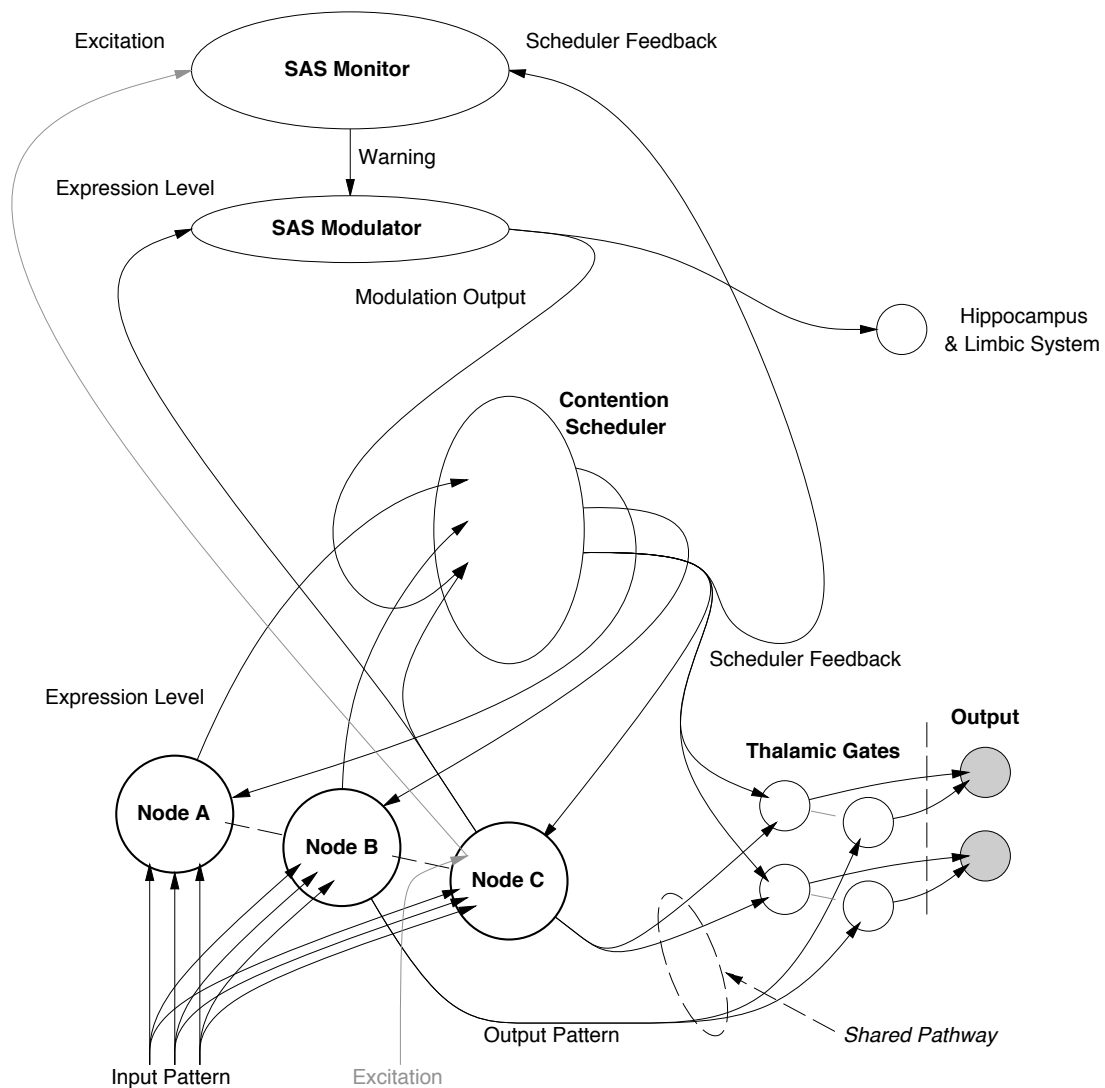


figure 5.11 Interconnections between a recogniser node and a SAS Monitor/Modulator node shown as a modified figure 5.2. The **Excitation** signal is generated from the rest of the (recurrent) network and may include priming inputs from other memory systems; the summed input of **Excitation** (to the recogniser node) is propagated to the SAS Monitor as its **Excitation** input signal. Node C's **Expression Level** and **Scheduler Feedback** signals are connected to the contention scheduler and replicated to the SAS Modulator and SAS Monitor respectively, as input signals. The SAS modulator's **Modulation Output** is connected to the contention scheduler for controlling access to the shared information pathway used by nodes B and C. Learning is achieved through the connection of **Modulation Output** to the hippocampus as previously described

Chapter 6

6. Implementation

This chapter describes the implementation of the attentional control architecture from the ideas developed in *Chapter 4* and *Chapter 5*. It begins by explaining how the elements of the model are specified and managed using XML and how this XML specification is used to construct the ‘runtime’ computational system (the Attentional Architecture). The chapter then proceeds to address the implementation of each of the elements in further detail, beginning with the basic building blocks of the control system, functional clusters of neurons, followed by each of the major functional modules in turn: episodic and procedural memory; the contention scheduler, including thalamic gating; working memory; the monitor and modulator elements of the Supervisory Attentional Systems; and, finally, elements of the limbic system which feature in adaptive learning, chiefly the hypothalamus. The chapter concludes with a description of how the functional neural clusters are trained together with the ‘runtime’ characteristics of the simulated control architecture.

6.1. Architectural Implementation

The Attentional Architecture is a large and complex entity. It is important that its implementation is transparent, that is, its construction must be examinable at an appropriate level; for this reason XML is used to specify the implemented architecture. In this chapter, reference is made to SNNS, a widely available (public domain) tool for specifying and training neural networks [Zell et al., 1993]. Detailed knowledge of SNNS is not required, but it is assumed that the reader is sufficiently familiar with the modelling and implementation of simple and common neural networks (e.g. Elman networks) and with standard approaches to training such networks, to allow them to interpret SNNS's specification notation. It is also assumed that the reader has an elementary appreciation of Extensible Markup Language (XML) [McLaughlin, 2000]. Finally, readers are assumed to have a basic familiarity with development of systems implemented in Java.

The main features of the implementation strategy for the Attentional Architecture are as follows:

- The elements of the system (its sub-systems and the connections or pathways between them) are specified using Extensible Markup Language (XML)
- Individual neural network files are specified by network definition files (currently SNNS formats are used)
- A defined naming convention, designed for the Attentional Architecture in Java 1.4, allows for composite networks defined in XML to be bound together
- Builders and Factories [Gamma et al, 1995] are used to 'instantiate' objects from XML files at runtime
- Rigourously enforced interfaces between all computationally functional elements of the system

6.1.1. Compositional Elements of the Control Network

An instance of the architecture is specified in a single 'controller definition file' (XML) which is used to build the computational control network (see *Appendix B* for the XML file used in *Chapter 7*). An overview of how the controlling architecture is composed is described below and illustrated in *figure 6.1*:

- **Clusters** are highly interconnected sets of neurons. Some clusters are pre-trained to perform particular functions (see below). Clusters are specified in 'cluster files' using the SNNS format for network definition files (see *listing 6.1*)
- **Networks** structure clusters. Several different cluster files can be used to build one functional network. In general there is a many to many relationship between **cluster files** and **networks**
- **Collections** structure **networks** into the major functional elements of the Attentional Architecture as presented in *Chapter 4*

- **Mappings** specify the pathways and connections between major elements of the architecture (networks and collections) by connecting sets of source neurons, to sets of target neurons. In order to implement the effects of neuromodulators, connections are ‘typed’ to support selective **chemosensitivity**, this allows neurons in any part of the architecture to be made sensitive to the simulated concentration of any neuromodulators present in the system

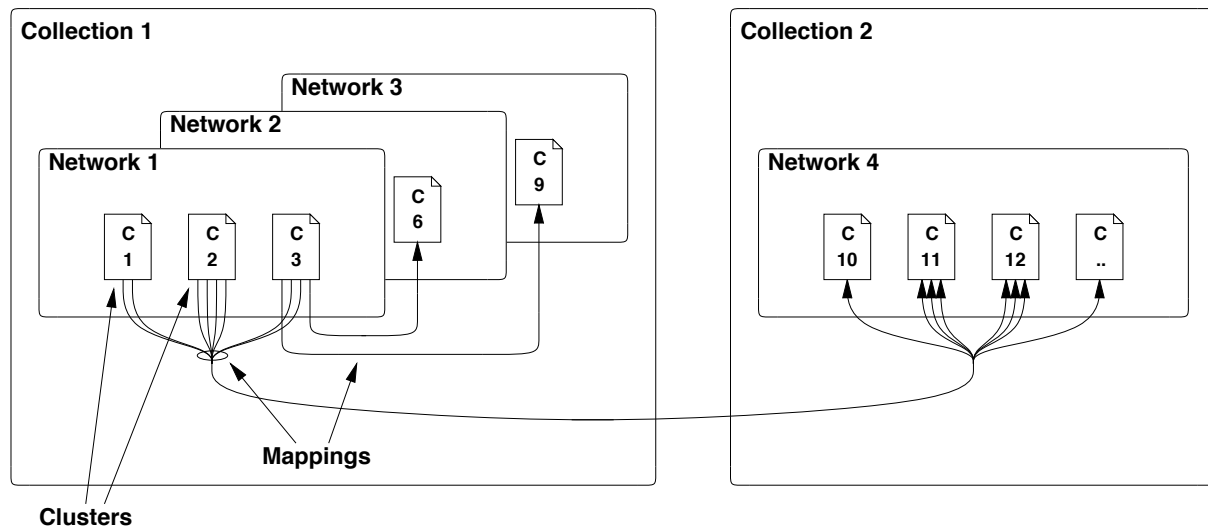


figure 6.1 An overview of the relationships between clusters (marked as **C1** etc), the networks that contain them, collections that structure networks and mappings that define the connections (pathways) between major architectural elements

6.1.2. Clusters

Within the architecture, basic functional components such as SAS Monitor/Modulator nodes or thalamic gateways are composed of ‘clusters’ of highly interconnected neurons specified in **cluster files**. The simulation code currently accepts cluster file definitions in SNNS format. A cluster may be composed of several (e.g. a thalamic gateway) to a few hundred neurons (e.g. a node in episodic memory).

The neuron types that make up these clusters are based on the standard neuron types found in SNNS. Each neural cluster within the architecture comprises a collection of highly interconnected neurons. Most are Elman or Jordan neural networks with four or eight inputs and up to three hidden layers. Both of these network topologies are recurrent so the state of the network is governed not just by its current input but also by its present state and all previous states and inputs, so behaviour of the system cannot be trivially predetermined but emerges as a result of each unique run of the system. The difference between the two networks is that in Elman networks the recurrent feedback is obtained from hidden layer neurons whilst in Jordan networks the feedback is obtained from the output layer.

Elman networks closely resemble the recognition clusters of Hinton, Plaut and Shallice, as specified in *Chapter 5, section 5.2*. Thus they feature extensively in the implementation of episodic and procedural

memory, were they were readily built and trained to categorise input patterns using the tools provided in SNNS. Jordan networks were used for highly specialised elements of the architecture (e.g. for the neural finite state machines that encode sequences of behaviours in working memory, see *section 6.2.4*), where it was necessary to be able to view and interpret the output of the network in order to undertake the 'hand tuning' of weights.

Each cluster is individually trained to a specification defined by the functional model in *Chapter 5*. Network requirements were identified, primarily through the use of timing diagrams (see *section 5.2.1.1*, *section 5.2.1.2* and *section 5.4*), component prototypes were then built, and pattern files generated either directly from the simulation or programatically (in Java). The resultant pattern files were then used to train (in SNNS) the network. Each cluster was tested independently and, if necessary, hand tuned. An example of a cluster file is given in *listing 6.1* and the associated cluster is illustrated in *figure 6.2*.

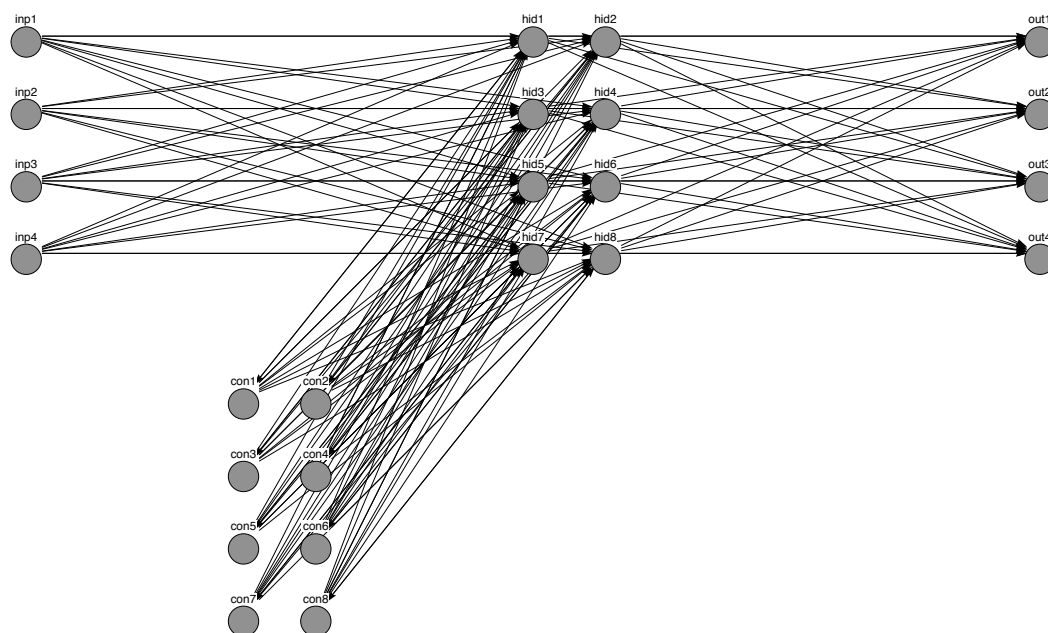


figure 6.2 A four input, four output Elman cluster, as shown in *listing 6.1* and displayed by SNNS; this particular network with two hidden layers, is a recognition cluster and forms part of a recogniser node in procedural memory, its role is to generate actuator signals (on out1-4) based on information routed from episodic memory (presented on inp1-4)

SNNS network definition file V1.4-3D
generated at Mon Oct 9 14:02:40 2006

network name : orient_chem_node
source files :
no. of units : 24
no. of connections : 144
no. of unit types : 0
no. of site types : 0

learning function : JE_BP
update function : JE_Order

unit default section :

act	bias	st	subnet	layer	act func	out func
0.00000	0.00000	h	0	1	Act_Logistic	Out_Identity

unit definition section :

no. sites	typeName	unitName	act	bias	st	position	act func	out func
1		inp1	0.09652	0.38200	i	1, 1,24	Act_Identity	
2		inp2	0.76246	0.85757	i	1, 2,24	Act_Identity	
3		inp3	0.28614	-0.79447	i	1, 3,24	Act_Identity	
4		inp4	0.05790	-0.59924	i	1, 4,24	Act_Identity	
5		hid1	0.00000	-1.20174	h	8, 1,24	Act_Logistic	
6		hid2	0.00547	-18.3663	h	9, 1,24	Act_Logistic	
...								
22		con6	0.00000	0.50000	h	5, 8,24	Act_Identity	
23		con7	0.00000	0.50000	h	4, 9,24	Act_Identity	
24		con8	0.00000	0.50000	h	5, 9,24	Act_Identity	

connection definition section :

target	site	source:weight
5		1:96.64687, 2:-96.30258, ..., 18: 0.42299, 19:-0.72549, 20: 0.43277
6		1:-4.76413, 2:12.88515, ..., 19: 0.31911, 20: 0.88728, 21:-0.12321
...		
23		11: 0.00000, 23: 0.00000
24		12: 0.00000, 24: 0.00000

Header information, used to verify the network built from each cluster file

The learning and update function used by SNNS for this cluster during training. (in this case Jordan-Elman back-propagation and ordering)

Unless otherwise specified in the *unit definition section* a neuron is given the default properties shown

Each neuron within the cluster file has a line in the *unit definition section*, specifying its properties i.e. its name, its bias, its activation function or its output function

Connections between neurons in the cluster file are specified in the *connection definition section*; target neurons are listed together with source neurons and connection weights (i.e. neuron 1 is connected to neuron 5 with a weight of 96.64687). A weight of 0.0 indicates the absence of a connection

listing 6.1 A typical SNNS cluster file as used in the Attentional Architecture, outlining the major sections (some detail has been omitted for clarity, shown by ...). These can be created by hand, programmatically (i.e. a Java procedure) or by SNNS graphical editors. A diagrammatical representation of this listing is shown in *figure 6.2* (above)

6.1.3. Networks

Networks group clusters. Within the simulation an unlimited number of duplicate networks can also be created; a number being automatically appended to the network name to ensure uniqueness. Within a network, neurons can be individually identified using the syntax [network name].[neuron name]. Creating a network from multiple clusters involves the use of the thread attribute; set to true the simulation will build a separate network for each cluster it creates from a cluster file, if set to false it will append the cluster to the last network it created. When referencing networks built using multiple clusters, the convention is to use a common prefix followed by “*” (e.g. the network created by *listing 6.2*, which contains three clusters would be referenced as `gate_*`) although the individual clusters can still be referenced directly (e.g. `gate_1`, `gate_2` etc.).

```
<!-- Three thalamic gates -->
<nuronet name="gate_1" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="true"/>
<nuronet name="gate_2" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="false"/>
<nuronet name="gate_3" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="false"/>
```

listing 6.2 Building the network `gate_*` from the SNNS cluster file `thalamic_gateway.net`. Note how setting the thread to false for `gate_2` and `gate_3` builds a single network of three clusters. The `dupl` attribute allows hundreds of networks to be built at once, with the network name being used as a prefix (i.e. `dupl="100"` would build a network with one hundred clusters named “`gate_1_1`” to “`gate_1_100`”; to access all the clusters in the network reference the name “`gate_1_*`”. Comments in XML are prefixed with `<!--` and end in `-->`.

Building a recogniser node (within episodic memory) is illustrated in *listing 6.3*. Each recogniser node is built from a control and a recognition cluster

```
<!-- A Recogniser Node -->
<nuronet name="control" cluster="control_cluster.net" type="SNNS" dupl="1" thread="true"/>
<nuronet name="recognition" cluster="recognition_cluster.net" type="SNNS" dupl="1"
thread="true"/>
```

listing 6.3 Building a fragment of episodic memory using recogniser nodes

6.1.4. Collections

Collections group networks (as defined above). Collections reflect the major functional elements of the gross architecture of the control system to be created. *Listing 6.4* illustrates a fragment of the XML specification of the thalamus, which itself comprises a number of collections, including a thalamus motor collection which in turn, comprises a number of thalamic gates which, subject to contention scheduling (see below), ‘gate’ signals from memory subsystems to robot drive motors. As with networks, individual neurons may be identified and monitored.

```

<nurocollection name="thalamus_gateway" type="collection" thread="true">
  <nurocollection name="motor" type="collection" thread="false">
    <nuronet name="gate_1" cluster=...
    <nuronet name="gate_2" cluster=...
    <nuronet name="gate_3" cluster=...
  </nurocollection>
</nurocollection>

```

listing 6.4 Defining a thalamus_gateway.motor collection

In procedural memory layers are collected together to form the basis of a hierarchy, multiple hierarchies as represented in *listing 6.5* (in this simple case one hierarchy is created for the gripper and one for the drive motors, this is then organised into "procedural_memory")

```

<nurocollection name="procedural_memory" type="collection" thread="true">
  <nurocollection name="motor" type="collection" thread="false">
    <nurocollection name="layer1" type="collection" thread="false">
      <nurocollection name="node1" type="collection" thread="false">
        <nuronet name="control" cluster=...
        <nuronet name="recognition" cluster=...
      </nurocollection>
      <nurocollection name="node2" type="collection" thread="false">
        <nuronet name="control" cluster=...
        <nuronet name="recognition" cluster=...
      </nurocollection>
    </nurocollection> <!-- End of Layer 1 -->
    <nurocollection name="layer2" type="collection" thread="false">
      ...
    </nurocollection> <!-- End of Layer 2 -->
  </nurocollection> <!-- End of motor hierarchy -->
  <nurocollection name="gripper" type="collection" thread="false">
    <nurocollection name="layer1" type="collection" thread="false">
      ...
    </nurocollection> <!-- End of Layer 1 -->
  </nurocollection> <!-- End of gripper hierarchy -->
</nurocollection>

```

listing 6.5 Using XML to define layers within procedural memory

6.1.5. Mappings

Mappings define the connections between elements of the architecture. Sets of 'source' neurons are mapped to sets of 'target' neurons, constituting pathways between architectural elements of the Attentional Architecture. They are built using a limited form of UNIX[®] regular expressions operating on the fully qualified names used in networks and collections. A source pattern is used to select a set of neurons that are 'mapped' to a set of neurons defined by the target pattern, as shown in *listing 6.6*. The document type definition (DTD) which outlines the syntax is shown in *Appendix B*. Connection weight (strengths) can also be given, (the default weighting is 1.0).

```

<nuromap source="contention_scheduler.motor.gate_1_enable"
  target="thalamus.motor.gate_1.enable" weight="1.0" type="1"/>

```

listing 6.6 Mapping a control output from the contention scheduler to an enable input in the motor thalamic gateway

Mappings can be used to create connections between any neurons but the convention is to only create connections between neurons in the input and output layers of the major architectural elements.

```
<!-- Map the outputs of node1 in episodic memory to node1 in procedural memory -->
<nuomap source="episodic_memory.layer1.node1.recognition.out*"
        target="procedural_memory.layer1.node1.recognition.inp#1" weight="1.0" type="1"/>

<!-- Map the salience output to the excitation input in episodic memory -->
<nuomap source="procedural_memory.layer1.node1.control.salience"
        target="episodic_memory.layer1.node2.control.excitation" weight="1.0" type="1"/>
```

listing 6.7 Creating information ‘pathway’ connections between episodic and procedural memory recogniser nodes and excitation/inhibitory connections between procedural and episodic memory recogniser nodes. Note: in the first mapping, the postfix ‘#’ is used to search the target set for *each* character match in the source set. (e.g. inp1 is mapped to out1, inp2 to out2, etc)

Connections between neurons can be rendered sensitive to fluctuations in concentration of simulated neuromodulatory chemicals the effect of which is to modulate the connection weights between neurons. Neuromodulation within the simulation is based on the observation by Fain [1999, p. 439] that there would appear to be little difference between metabotropic transmission (neuromodulation) between neurons and ionotropic communication; both often use the same transmitters. Hence neurons within the simulation have a refractory period after a neuromodulatory stimulus has triggered a response in which no other neuromodulator will be responded to. Connection types within the Attentional Architecture are analogous to synapses created by ‘connexins’ [Fain, 1999, pp. 256-257], these create the gap junctions between neurons with different connexin proteins having different electrical properties. There is no support for ‘typed’ connections within SNNS so the type attribute can only be used with mappings created using the `nuomap` element. Within the model the default type is “1”; neurons which only have type 1 connections are ‘ordinary’ neurons (i.e. the output is a function of the sum of the inputs). Neurons containing type “2” connections are sensitive to neuromodulators (i.e. the output is the product of a factor representing the response to neuromodulation, itself a function of the chemosensitivity threshold and the chemorefractory state of the neuron in addition to the function of the input summation). Only inhibitory links between recogniser nodes are type “2” in the current architecture, this allows for their modulation in the presence of a neuromodulator described below. *Listing 6.8* illustrates a specification in which the neurons of the pathway (mapping) are ‘sensitized’ to neuromodulator “1” with a sensitivity threshold of 0.70 and a refractory period of 80. *Listing 6.9* shows a similar specification, sensitising “type 2” inhibitory links between recogniser nodes in episodic memory.

```
<sensitize modulator ="1" threshold="0.70" refractory="80">
  <nuomap source="contention_scheduler.motor.gate_1_enable"
    target="thalamus.motor.gate_1.enable" weight="1.0" type="1"/>
</sensitize>
```

listing 6.8 Using a “sensitize” element to allow neuromodulation of the mapping previously shown in *listing 6.6*. All the ‘target’ neurons selected by the mapping (nuomap) will automatically be sensitized to the neuromodulator specified by the “modulator” attribute. Several “sensitize” commands can be active at once.

```
<sensitize modulator ="1" threshold="0.70" refractory="80">
  <nuomap source="episodic_memory.layer2.node*.control.saliency"
    target="episodic_memory.layer1.node*.control.excitation" weight="-0.1" type="2"/>
</sensitize>
```

listing 6.9 Using a “sensitize” element to allow neuromodulation of inhibitory connections between episodic recogniser nodes

6.2. *Developing Major Functional Elements of the Architecture*

6.2.1. *The Controlling Network*

The currently implemented version of the Attentional Architecture (see *section 7.2*) is made up of sixty three neural clusters organised into the functional modules of episodic memory, working memory, procedural memory, contention scheduler, SAS, hippocampus (limbic system) and thalamus that correspond to the architecture described in *figure 4.2* from *Chapter 4*. As implemented the controlling network is a restricted version of the architecture defined in *figures 4.8, 4.9* and *4.10* in that there is no SAS Generator and the memory hierarchies are very small.

6.2.2. *Episodic Memory and Procedural Memory*

Episodic memory’s primary purpose is to analyse and record the machines interactions with the environment. Procedural memory has an equivalent role in generating actions based on the analysis and activity of episodic memory. There is some similarity between these structures:

- Both episodic memory and procedural memory are hierarchically organised, lower layers are connected to sensors and actuators respectively, whilst higher layers are connected (in increasing numbers) to working memory
- Both have several hierarchies [Fuster, 2003], episodic memory has a separate hierarchy for each sensory mode (i.e. touch, smell etc.), whilst procedural memory has a separate hierarchy for major anatomical components (i.e. arms, legs etc.)
- Episodic memory is heavily interconnected via ‘information pathways’ to procedural memory (Connection A in *figure 6.3*), which uses the information to achieve ‘reactive’ coupling between perception and action. Reciprocal links exist from procedural memory to episodic memory [Frith, 1998] providing the machine with anticipated results of its actions (*priming*)

Episodic memory is composed of recogniser nodes organised into layers, each layer is part of a hierarchical structure and there are multiple hierarchies, each corresponding to a class of sensory input, increasingly fused towards the top. Nodes at the bottom of each hierarchy interface to the perception layers and represent 'atomic' episodes (e.g. food is present, home is present); nodes further up each hierarchy represent increasingly compound episodes. Activation is propagated up the episodic memory hierarchical structure and across to equivalent layers in procedural memory (priming). Within the current architecture episodic memory is primarily concerned with fusing atomic episodes (e.g. food is present, the gripper is holding food and home is present therefore food may be taken home).

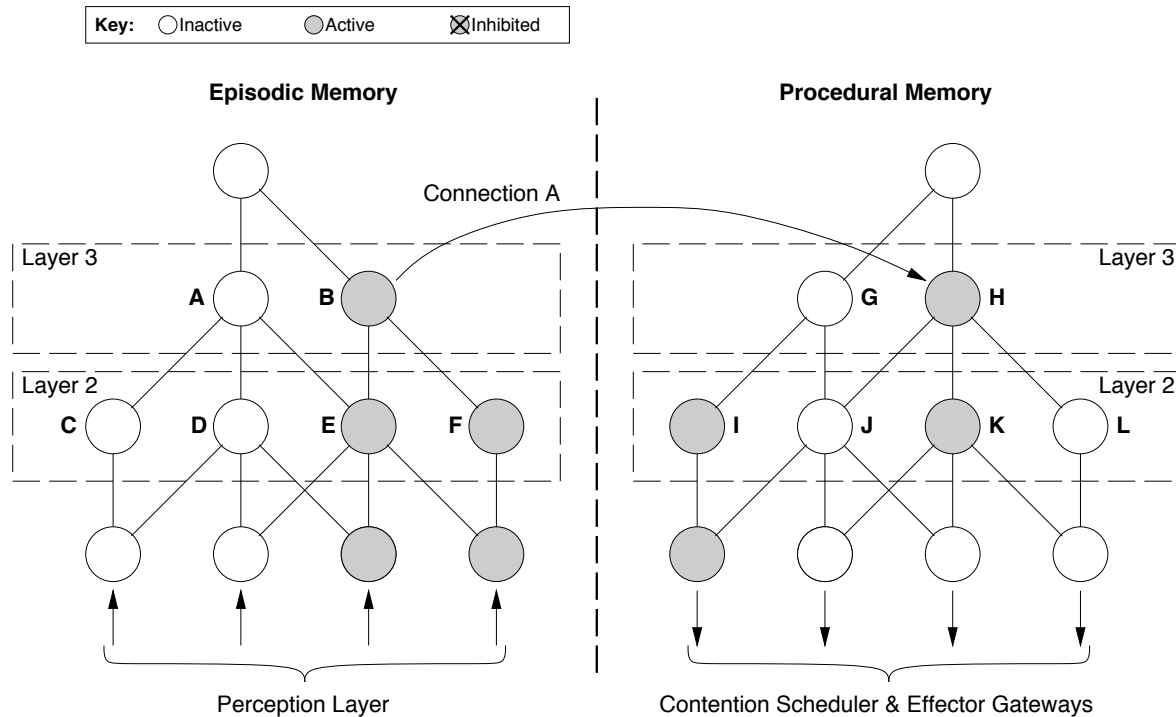


figure 6.3 The structure of the reactive links of episodic and procedural memory and their interconnections. Perception induces a pattern of excitation in episodic memory, this in turn excites nodes at corresponding levels in procedural memory, this then propagates activation to nodes representing 'atomic' actions. At each level in episodic and procedural memory nodes may have to compete for expression through the contention scheduler. (Note: mutually inhibitory links between layered nodes in episodic and procedural memory respectively are not shown)

Procedural memory recogniser nodes are layered in a similar fashion to episodic memory, again with multiple hierarchies representing major actuator classes (e.g. an arm) increasingly fused at the top; nodes at the bottom of each hierarchy (*pattern generators*) represent primitive motor actions (e.g. close gripper), with nodes further up the hierarchy (*top down recogniser nodes*) representing compound motor actions (e.g. orientate towards home and then move forward). Activation is propagated from higher to lower levels in procedural memory and across to equivalent layers in episodic memory (priming).

```

<nurocollection name="episodic_memory" type="collection" thread="true">
  <nurocollection name="layer2" type="collection" thread="true">
    <nurocollection name="node_c" type="collection" thread="false">
      <nurocollection name="control" cluster="control_cluster.net" type="SNNS" dupl="1"
        thread="false"/>
      <nurocollection name="recognition" cluster="recognition_cluster.net" type="SNNS" dupl="1"
        thread="false"/>
    </nurocollection>
    <nurocollection name="node_d" ...
  </nurocollection>
    <nurocollection name="node_e" ...
  </nurocollection>
    <nurocollection name="node_f" ...
  </nurocollection>
    </nurocollection>
  <nurocollection name="layer3" type="collection" thread="true">
    <nurocollection name="node_a" ...
  </nurocollection>
    <nurocollection name="node_b" ...
  </nurocollection>
  </nurocollection>
</nurocollection>

<nurocollection name="procedural_memory" type="collection" thread="true">
  <nurocollection name="layer2" type="collection" thread="true">
    <nurocollection name="node_i" ...
  </nurocollection>
    <nurocollection name="node_j" ...
  </nurocollection>
    <nurocollection name="node_k" ...
  </nurocollection>
    <nurocollection name="node_l" ...
  </nurocollection>
  </nurocollection>

  <nurocollection name="layer3" type="collection" thread="true">
    <nurocollection name="node_g" ...
  </nurocollection>
    <nurocollection name="node_h" ...
  </nurocollection>
  </nurocollection>
</nurocollection>

<!-- Map Episodic Memory, salience outputs of Layer 2 into the excitation inputs of Layer 3 -->
<!-- Excitation flow in episodic memory is bottom up -->
<!-- nurocollections can be used to group mappings for readability -->
<nurocollection name="episodic_memory_feed_forward" type="nuromapping" thread="false">
  <nuromap source="episodic_memory.layer2.node*.control.salience"
    target="episodic_memory.layer3.node*.control.excitation" weight="1.0">
  </nuromap>
</nurocollection>

<!-- Map Procedural Memory, salience outputs of layer 3 into the excitation inputs of layer 2 -->
<!-- Excitation flow in procedural memory is top down -->
<nurocollection name="procedural_memory_feed_forward" type="nuromapping" thread="false">
  <nuromap source="procedural_memory.layer3.node*.control.salience"
    target="episodic_memory.layer2.node*.control.excitation" weight="1.0">
  </nuromap>
</nurocollection>

<!-- Map a priming link from Episodic to Procedural Memory -->
<nurocollection name="episodic_priming_links" type="nuromapping" thread="false">
  <!-- Connection A -->
  <nuromap source="episodic_memory.layer3.node_b.control.salience"
    target="procedural_memory.layer3.node_h.control.excitation" weight="1.0">
  </nuromap>
</nurocollection>

```

listing 6.10 The XML for implementing a fragment of the episodic and procedural memory structures shown in *figure 6.3*

In both episodic and working memory inhibitory and excitatory links between adjacent layers (and hierarchies) ensure that salient nodes can excite (prime) their sub-nodes, causing them to respond more readily when the conditions for their activation arise. In this way, episodic memory exhibits priming in respect of anticipated perceptual input and procedural memory exhibits priming of the relevant top-down recogniser node. In this implementation, the combination of episodic and

procedural memory realises a small number of reactive basis behaviours (c.f. Mataric [1996]). Basis behaviours are low-level behaviours that may be combined to provide higher-level compound behaviours. The basis behaviours, and higher-level behaviours arising from them, serve a similar role as “schemata” in the Norman & Shallice model. Other representations of schemata exist such as those of by Cooper [2003] and Kasderidis & Taylor [2004]. Most recogniser nodes (episodic and procedural) have inputs from other recogniser nodes and feedback from the contention scheduler (see below) where it competes for access to information paths (e.g. working memory or effectors). The output of each memory cluster to the contention scheduler (expression level) represents the strength of a request for expression of the memory (its net salience).

6.2.3. The Contention Scheduler and Thalamus

Within the model, the contention scheduler is based on the computational properties of the basal ganglia [Baddeley, 1987], [Houk et al., 1995] and is an independent implementation of the contention scheduler described by Prescott et al. [1999]. A four input (and output) winner takes all cluster is used repeatedly throughout the contention scheduler for episodic, procedural and working memory; each input has an associated input used by the SAS to modulate its operation.

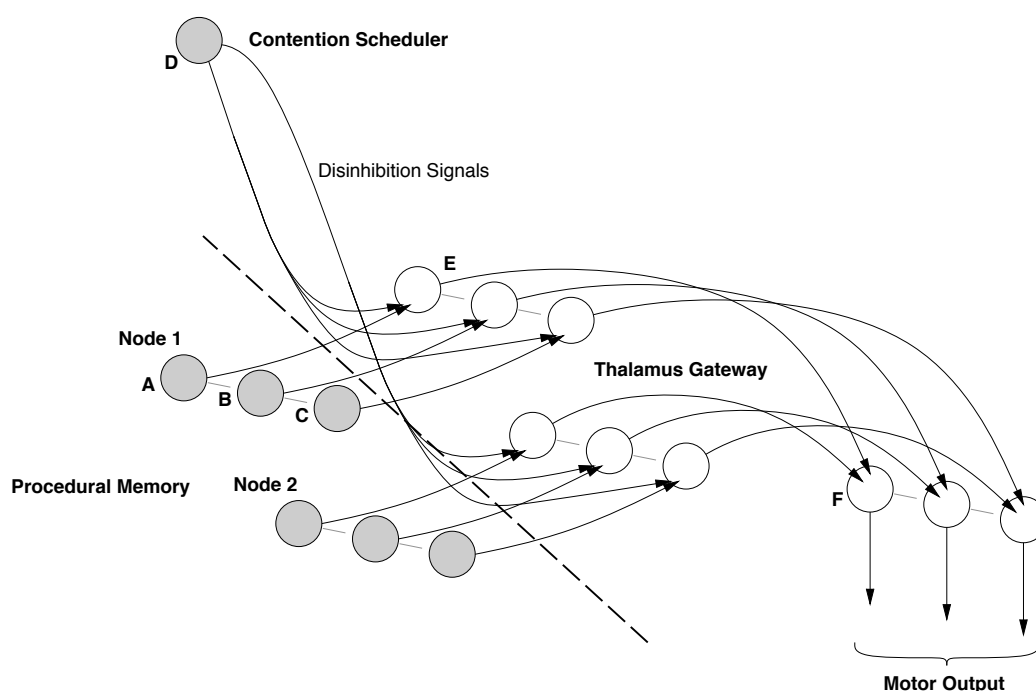


figure 6.4 An overview of how the contention scheduler, thalamus gateways and memory subsystems interact. In this diagram two nodes in procedural memory (1 and 2) are competing for access to the motor actuators. Node 1 has several output neurons (A, B & C) that are connected into the thalamus (i.e. neuron A of node 1 is connected to gateway E). Unless the contention scheduler ‘disinhibits’ a particular node’s output, it is by default, inhibited. The control networks of each node are not shown (e.g. the connection from node 1 to the contention scheduler D), however the ‘disinhibition’ signals from the contention scheduler, to the thalamus gateway are. If node 1 wins the competition to be expressed, the contention scheduler disinhibits the corresponding thalamic gates allowing the node access (e.g. neuron A is allowed access to node F via thalamic gate E)

```

<nurocollection name="procedural_memory" type="collection" thread="true">
  <nurocollection name="layer2" type="collection" thread="true">
    <nurocollection name="node_1" type="collection" thread="false">
      <nuronet name="control" cluster="control_cluster.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuronet name="recognition" cluster="recognition_cluster.net" ...
    </nurocollection>
    <nurocollection name="node_2" type="collection" thread="false">
      <nuronet name="control" cluster="control_cluster.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuronet name="recognition" cluster="recognition_cluster.net" ...
    </nurocollection>
  </nurocollection>
</nurocollection>

<nurocollection name="thalamus" type="collection" thread="true">
  <nurocollection name="procedural_memory" type="collection" thread="false">
    <nurocollection name="node_1" type="collection" thread="false">
      <nuronet name="gate1" network="thalamic_gateway.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuronet name="gate2" network="thalamic_gateway.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuronet name="gate3" network="thalamic_gateway.net" type="SNNS" dupl="1"
        thread="false"/>
    </nurocollection>
    <nurocollection name="node_2" type="collection" thread="false">
      <nuronet name="gate1" network="thalamic_gateway.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuronet name="gate2" network="thalamic_gateway.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuronet name="gate3" network="thalamic_gateway.net" type="SNNS" dupl="1"
        thread="false"/>
    </nurocollection>
  </nurocollection>
</nurocollection>

<nurocollection name="contention_scheduler" type="collection" thread="true">
  <nurocollection name="procedural_memory" type="collection" thread="false">
    <nuronet name="layer2" network="contention_scheduler.net" type="SNNS" dupl="1" thread="false"/>
  </nurocollection>
</nurocollection>

<nuronet name="motor_output" network="motor_output.net" type="SNNS" dupl="1" thread="false">

<!-- Map the procedural memory outputs into the thalamic gates -->
<nuromap source="procedural_memory.layer2.node_1.recognition.outA"
  target="thalamus.procedural_memory.node_1.gate1.inp2" weight="1" type="1"/>
<nuromap source="procedural_memory.layer2.node_1.recognition.outB"
  target="thalamus.procedural_memory.node_1.gate2.inp2" weight="1" type="1"/>
<nuromap source="procedural_memory.layer2.node_1.recognition.outC"
  target="thalamus.procedural_memory.node_1.gate3.inp2" weight="1" type="1"/>

<nuromap source="procedural_memory.layer2.node_2.recognition.outA"
  target="thalamus.procedural_memory.node_1.gate1.inp2" weight="1" type="1"/>
<nuromap source="procedural_memory.layer2.node_2.recognition.outB"
  target="thalamus.procedural_memory.node_1.gate2.inp2" weight="1" type="1"/>
<nuromap source="procedural_memory.layer2.node_2.recognition.outC"
  target="thalamus.procedural_memory.node_1.gate3.inp2" weight="1" type="1"/>

<!-- Map the Expression Level output of each node into the contention scheduler -->
<nuromap source="procedural_memory.layer2.node_1.control.expression_level"
  target="contention_scheduler.procedural_memory.layer2.inp1" weight="1" type="1"/>
<nuromap source="procedural_memory.layer2.node_2.control.expression_level"
  target="contention_scheduler.procedural_memory.layer2.inp2" weight="1" type="1"/>

<!-- Map the output of the contention scheduler into the thalamic gates disinhibition inputs -->
<nuromap source="contention_scheduler.procedural_memory.layer2.out1"
  target="thalamus.procedural_memory.node_1.gate*.inp1" weight="1" type="1"/>
<nuromap source="contention_scheduler.procedural_memory.layer2.out2"
  target="thalamus.procedural_memory.node_2.gate*.inp1" weight="1" type="1"/>

<!-- Map the combined output of the thalamic gates to the motors -->
<nuromap source="thalamus.procedural_memory.node*.gate1.out1"
  target="motor_output.neuronF" weight="1" type="1"/>
<nuromap source="thalamus.procedural_memory.node*.gate2.out1"
  target="motor_output.neuronG" weight="1" type="1"/>
<nuromap source="thalamus.procedural_memory.node*.gate3.out1"
  target="motor_output.neuronH" weight="1" type="1"/>

```

listing 6.11 The XML for implementing the structures illustrated in *figure 6.4*

The basic building blocks of the thalamus [Mumford, 1999; Gurney et al, 2001] are the ‘thalamic gates’ used in the implementation. An illustration of the neural composition of a single ‘thalamic gate’ is displayed in *figure 6.5* with one of these gates corresponding to the node marked **E** within *figure 6.4*. The output of a thalamic gate is inhibited by default, it has to be actively disinhibited by a contention scheduler for it to become enabled. Each shared information pathway is protected from inappropriate access by several of these gateways as shown in *figure 6.4*.

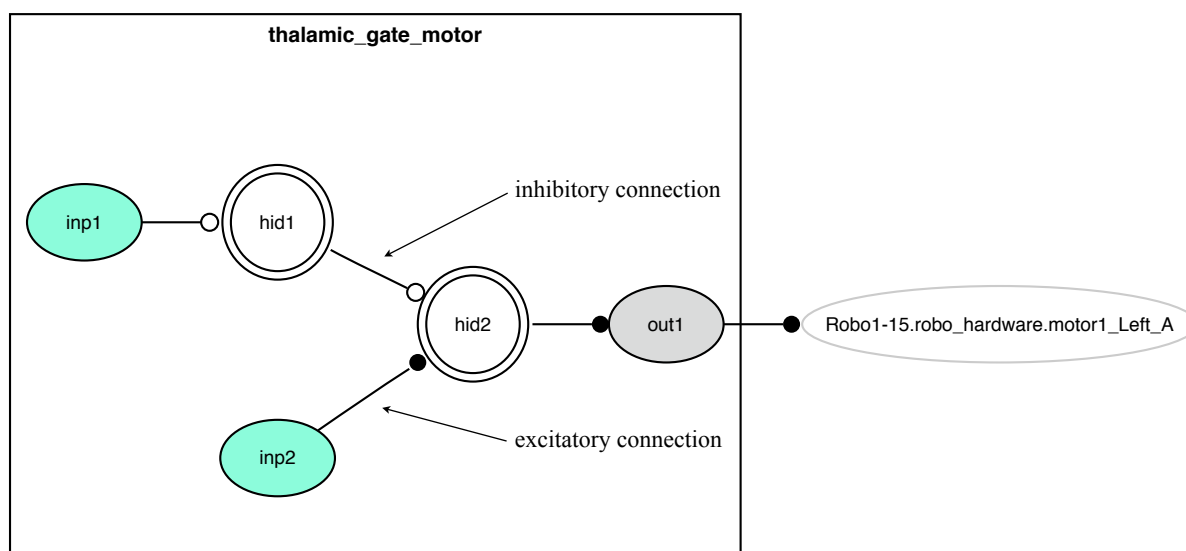


figure 6.5 The ‘thalamic gates’ used extensively in the thalamus, protect shared information pathways from inappropriate access by competing recogniser nodes. This gate is shown as the node marked **E** on *figure 6.4*. Information from the memory subsystem (node 1, **A** on *figure 6.4*) is presented on *inp2* and if the contention scheduler has disinhibited the gate (shown as the link from the contention scheduler to node **E** on *figure 6.4* and *inp1* on *figure 6.5*) will cause this information to be replicated to *out1*, shown in this diagram connected to a motor actuator neuron (shown in *figure 6.4* as node **F**)

6.2.4. Working Memory

The role of working memory is to organise and sequence behaviour in time, so that an overall goal based performance target (set externally or by the SAS Generator) is achieved. In ill-learned situations Norman & Shallice propose that the SAS and working memory are heavily involved in the internal sequencing of behaviour and/or actions and hence express *willed intention*.

Working memory encodes the salience of high-level nodes in episodic and procedural memory, providing the executive with a representation of how perception is currently related to action at a reactive level. Recogniser (observer) nodes within working memory also encode the currently intended task so that the SAS monitor can detect any discrepancy between intention and reactive action. As previously outlined in *figure 4.4* and *figure 4.5* from *Chapter 4*, its structure is similar in structure to both episodic and indirectly, procedural memory, however it would not appear to be somatotrophically organised.

Hence, in order to facilitate the sequencing of sub-behaviours, recogniser nodes behave as neural finite state machines; changes in inputs trigger a state dependent transition to a next state and its associated output. The timing of the sequence of behaviours is not pre-determined as the exact state of the network is a function of the current state of the environment, the history of the machine and any executive attentional activity.

```
<nurocollection name="working_memory" type="collection" thread="true">
  <nurocollection name="layer1" type="collection" thread="true">
    <nurocollection name="node_a" type="collection" thread="false">
      <nuuronet name="control" cluster="control_cluster.net" type="SNNS" dupl="1"
        thread="false"/>
      <nuuronet name="recognition" cluster="recognition_cluster.net" type="SNNS" dupl="1"
        thread="false"/>
    </nurocollection>
    <nurocollection name="node_b" ...
  </nurocollection>
</nurocollection>

<nurocollection name="layer2" type="collection" thread="true">
  <nurocollection name="node_c" type="collection" thread="false">
    <nuuronet name="control" cluster="control_cluster.net" type="SNNS" dupl="1"
      thread="false"/>
    <nuuronet name="recognition" cluster="recognition_cluster.net" type="SNNS" dupl="1"
      thread="false"/>
  </nurocollection>
  <nurocollection name="node_d" ...
</nurocollection>
  <nurocollection name="node_e" ...
</nurocollection>
  <nurocollection name="node_f" ...
</nurocollection>
</nurocollection>

<!-- mappings to node f -->
<nuomap source="working_memory.layer1.node_a.control.salience"
  target="working_memory.layer2.node_f.recognition.inp1" weight="1" type="1"/>
<nuomap source="working_memory.layer1.node_b.control.salience"
  target="working_memory.layer2.node_f.recognition.inp2" weight="1" type="1"/>
<nuomap source="working_memory.layer2.node_d.recognition.out1"
  target="working_memory.layer2.node_f.recognition.inp3" weight="1" type="1"/>
<nuomap source="working_memory.layer2.node_d.recognition.out2"
  target="working_memory.layer2.node_f.recognition.inp4" weight="1" type="1"/>
<nuomap source="working_memory.layer2.node_d.recognition.out3"
  target="working_memory.layer2.node_f.recognition.inp5" weight="1" type="1"/>

<nuomap source="working_memory.layer2.node_d.control.salience"
  target="working_memory.layer2.node_f.control.excitation" weight="1" type="1"/>
```

listing 6.12 Working memory including mappings as implemented using XML, based on *Chapter 5, figure 5.7*. For clarity of reading the XML, the default weight of 1.0 is always shown

In the implementation of working memory, the activity of the SAS Generator (the action of which sequences and plans behaviours in working memory) is simulated by using one of the neural finite state machines described above.

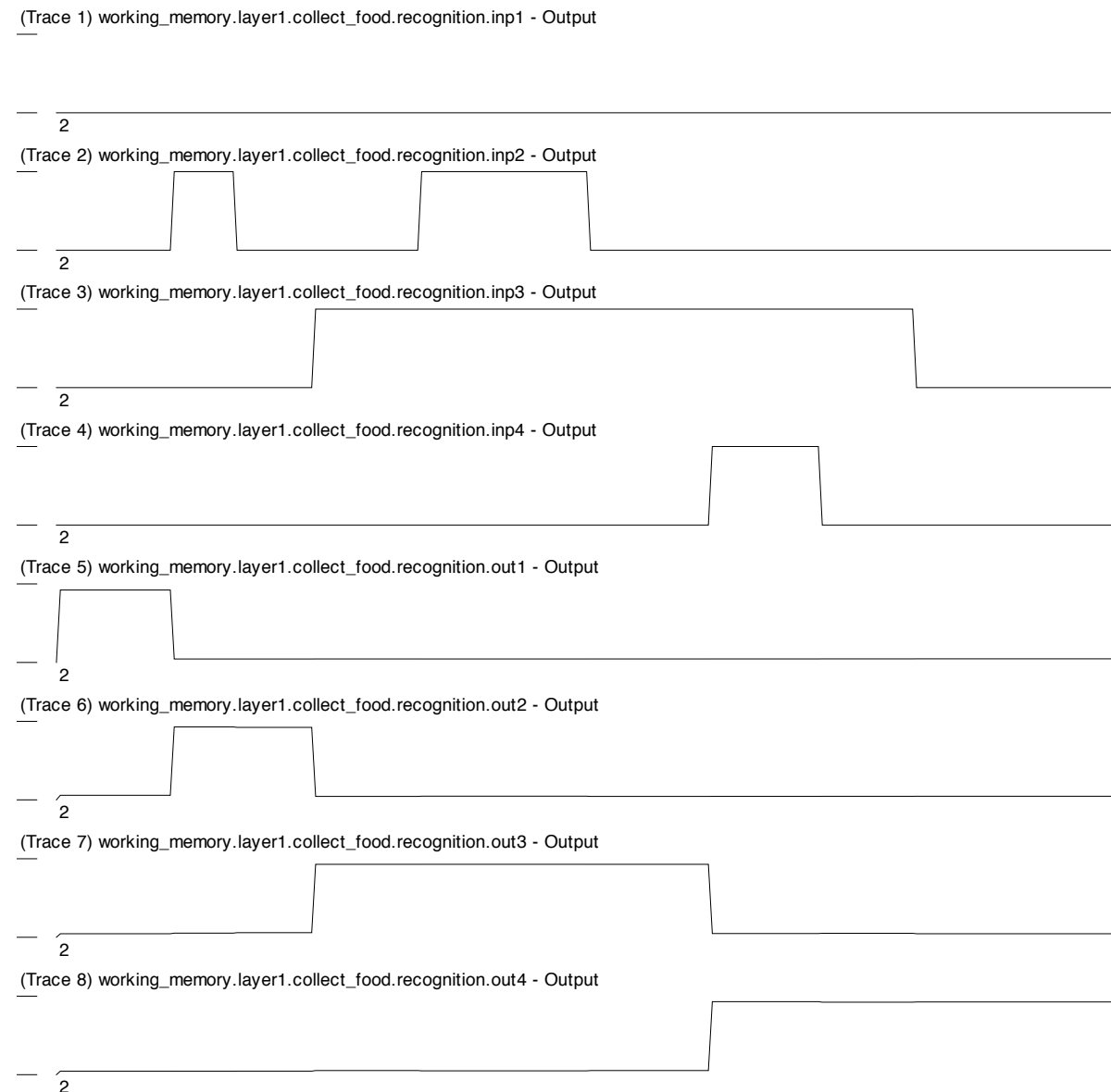


figure 6.6 The simulation traces of the working memory node 'collect_food' taken from *SheepWorld3*, showing the willed intention of the machine as it generates a sequence of behaviours. The input pattern from episodic memory is shown on inputs one to four (inp1, **Trace 1** to inp4, **Trace 4**) whilst the output pattern of the node (connected to procedural memory) is shown as outputs one to four (out1, **Trace 5** to out4, **Trace 8**). The initial state of this particular node is to always have an 'intention', that is, at the start of the sequence output one will always be high (the node can be reset to its initial state by an input not shown on the trace, to allow for repetitive cycles of activity, interruption etc) as can be seen from **Trace 5**. The rising edge of input two (inp2, **Trace 2**) causes the node to move to the next sequence, shown as the rising edge of output two (out2, **Trace 6**) and the falling edge of output one (out1, **Trace 1**); neither the first falling edge of input two nor any subsequent activity by input two has any effect on the node. The rising edge of input three (inp3, **Trace 3**) causes the transition to the next state as seen by the rising edge of output three (out3, **Trace 7**) and corresponding falling edge of output 2; similarly the rising edge of input four (inp4, **Trace 4**) causes the rising edge of output four (out4, **Trace 8**) and falling edge of output three.

6.2.5. The Supervisory Attentional System

As already outlined, the SAS has several functions; currently two are implemented: Monitor and Modulate, described in *Chapter 5* (see *figure 6.7*). The full SAS has a Generate function to create novel plans but this function has not yet been implemented; instead the output of the SAS Generator is simulated as an encoding of sequences of intended tasks (i.e. plans) in working memory. Thus, human intervention in the normal operation of the controller is confined to specifying at the outset the intended goal of the robot, encoded as a neural finite state machine (see *section 6.2.4*, above).

In the current implementation each recogniser node that can be modulated in episodic, working and procedural memory has a corresponding monitor and modulator node. Each SAS monitor node receives input from three sources:

- the summed **Excitation** signals to the recogniser node, which expresses the summed intention (the plan) of the entire system to activate that node
- **Expression Level**, the signal from the recogniser node to the contention scheduler(s) to request the expression of its output pattern
- the **Scheduler Feedback** signal from the contention scheduler to the nodes i.e. the reactive response of the system

The SAS Monitor raises a warning stimulus to the SAS Modulator if a node is inappropriately expressed. Once triggered, the warning signal continues to be generated until the conflict is resolved by changes in the relative salience levels of **Excitation**, **Expression Level** and **Scheduler Feedback** inputs.

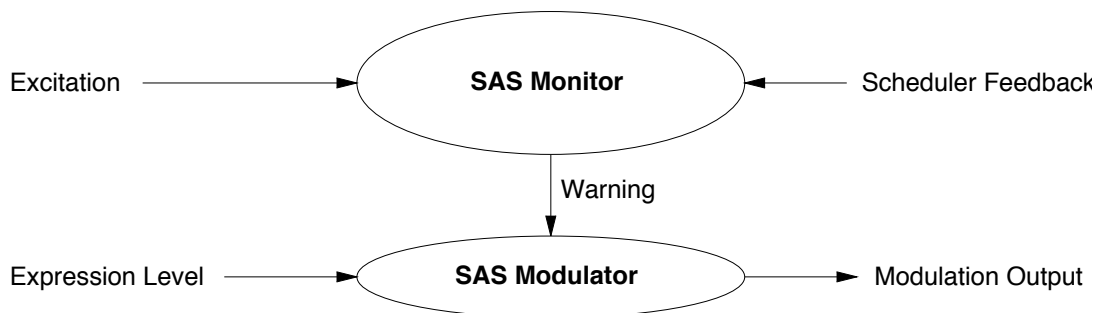


figure 6.7 An overview of a SAS Monitor and SAS Modulator from *Chapter 5*

Illustrated in *figure 6.8* is a representation of the neurons that make up the SAS monitor and modulator for the working memory recogniser (observer) node 'collect_food'. This encodes the mappings between episodic and procedural memory to find, pick up and carry food home; it is used as the basic behaviour for the validation experiments in *Chapter 7*. *Listing 6.13* illustrates how the SAS is built and connected into episodic memory. Its signal traces, taken from SheepWorld3 are shown in *figure 6.9*.

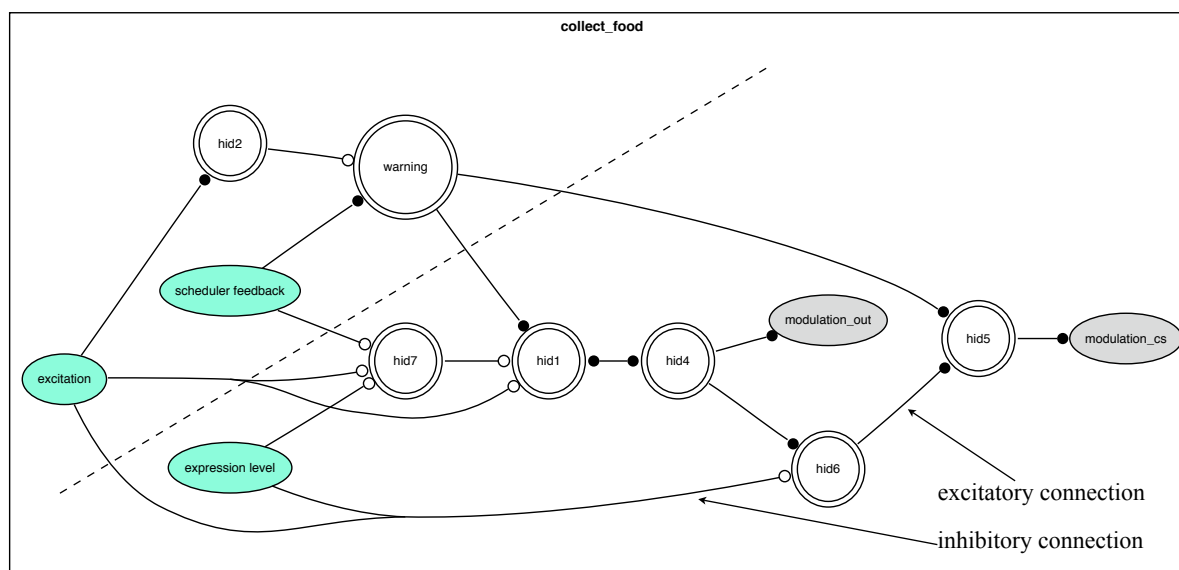


figure 6.8 The SAS monitor and modulator for the working memory observer node named 'collect_food'. Although it is implemented as a single structure, the monitor and modulator components are marked on the diagram. The output marked `modulation_cs` provides the suppression input into the contention scheduler, whilst `modulation_out` provides input into the limbic structures. Neurons internal to the cluster are shown with double circles, other neurons are named the same as their external input/output signals. The neuron labelled 'warning' provides the arousal signal to the modulator. The SAS clusters were hand built for the Attentional Architecture and visualized using *Graphvis*

SAS Modulation clusters generate output patterns that modulate the signals from nodes within the memory subsystems into the contention scheduler so that intended behaviours (episodes/actions) are potentiated and other behaviours (competing for expression) are attenuated. It is important to recognise that this does not guarantee the selection of the intended behaviour, as this risks overriding behaviours strongly and appropriately triggered by the environment, e.g., those designed to prevent harm to the machine or its surroundings. Each recogniser node in episodic, working and procedural memory that uses a contention scheduler within the domain of the SAS (as outlined *Chapter 5, figure 5.10*) has an associated SAS Modulation cluster that is independent of the specific behaviour the node is part of.

```

<nurocollection name="episodic_memory" type="collection" thread="true">
    <nurocollection name="layer3" type="collection" thread="true">
        <nurocollection name="node_a" type="collection" thread="false">
            <nuronet name="control" cluster="control_cluster.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" cluster="recognition_cluster.net" type="SNNS" dupl="1"
                thread="false"/>
        </nurocollection>
        <nurocollection name="node_b" ...
    </nurocollection>
</nurocollection>

<nurocollection name="sas" type="collection" thread="true">
    <nurocollection name="monitor_modulate" type="collection" thread="false">
        <nurocollection name="episodic_memory" type="collection" thread="false">
            <nurocollection name="layer3" type="collection" thread="false">
                <nuronet name="node_a" cluster="sas_monitor_modulator.net" type="SNNS" dupl="1"
                    thread="false"/>
            </nurocollection>
        </nurocollection>
    </nurocollection>
</nurocollection>

<!-- Excitation (Input) -->
<nuromap source="episodic_memory.layer3.node1.control.excitation"
    target="sas.monitor_modulate.episodic_memory.layer3.node_a.excitation" weight="1" type="1"/>

<!-- Expression Level (Input) -->
<nuromap source="episodic_memory.layer3.node1.control.expression_level"
    target="sas.monitor_modulate.episodic_memory.layer3.node_a.expression_level" weight="1" type="1"/>

<!-- Scheduler Feedback (Input) -->
<nuromap source="contention_scheduler.episodic_memory.layer3.out1"
    target="sas.monitor_modulate.episodic_memory.layer3.node_a.scheduler_feedback" weight="1" type="1"/>

<!-- Modulation CS (Output to the Contention Scheduler's Suppression Input) -->
<nuromap source="sas.monitor_modulate.episodic_memory.layer3.node_a.modulation_cs"
    target="contention_scheduler.episodic_memory.layer3.suppression_input1" weight="1" type="1"/>

<!-- Modulation Out (Output to the Hippocampus) -->
<nuromap source="sas.monitor_modulate.episodic_memory.layer3.node_a.modulation_out"
    target="hippocampus.episodic_memory.layer3.node_a.input1" weight="1" type="1"/>

```

listing 6.13 The XML to create the SAS Monitor and Modulation cluster shown in *figure 6.8*, together with its associated mappings connecting to and from the recogniser node **B** from *figure 6.4*

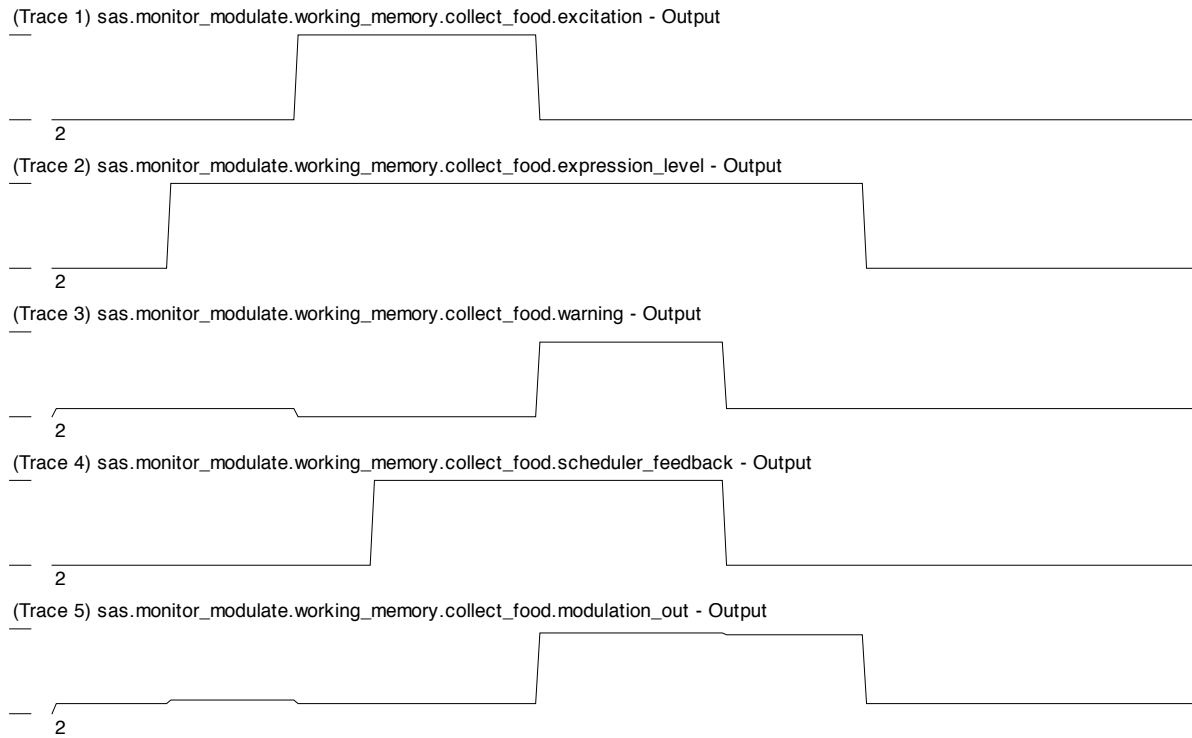


figure 6.9 A simulation trace of the SAS cluster associated with the working memory node 'collect_food' generated by the SheepWorld3 Navigator panel, validating the operation of a SAS cluster as required in Chapter 5, section 5.4. To generate this trace the SAS inputs were forced 'on' as in figure 5.9 and the resulting warning and modulation_out signals captured. The number (2) underneath each trace shows the global clock (the actual simulation was suspended during this capture), in normal operation second 'ticks' accompany the trace output so that simulation events and internal traces can be correlated after the simulation ends. The rising edge of **Expression Level** (expression_level, **Trace 2**) indicates that the recogniser node that the SAS node is monitoring has become active (i.e. it has recognised a pattern on its inputs and is requesting expression through a contention scheduler), though it is currently unplanned, the SAS ignores this input until the rising edge of **Excitation** (excitation, **Trace 1**) together with at some point its scheduler feedback (the rising edge of **Scheduler Feedback**, scheduler_feedback, **Trace 4**). After expression the recogniser node has completed its task within the behaviour and **Excitation** falls (**Trace 1**). As the activity of the recogniser node is now no longer planned, the SAS monitor activates the **Warning** signal (**Trace 3**) causing the SAS Modulator to become excited (the rising edge of modulation_out, **Trace 5**). Due to the modulation of the contention scheduler, the induced learning caused by the modulation or the environment, the recogniser node becomes deselected by the contention scheduler and loses expression (the falling edge of **Trace 4**) at the same time as the **Warning** signal becomes inactive (falling edge of **Trace 3**); however the recogniser node is still activated, the **SAS Modulation Output** (**Trace 5**) continues its activity until the falling edge of **Expression Level** (**Trace 2**) is detected

6.2.6. The Affective System, Hippocampus and Learning

The overall level of activity of the SAS modulator provides the basis for a signal to the affective system. Low levels of activity in the SAS Modulator are associated with routine (minimally-attended) responses to familiar circumstances; high levels of activity reflect attentional effort in the face of unfamiliar circumstances. As described above, each recogniser node in memory has inhibitory and excitatory connections from neighbouring nodes. The nodes are sensitive to neuromodulatory chemicals (released by the limbic system), which have different effects based on the neurons' activities and connection types.

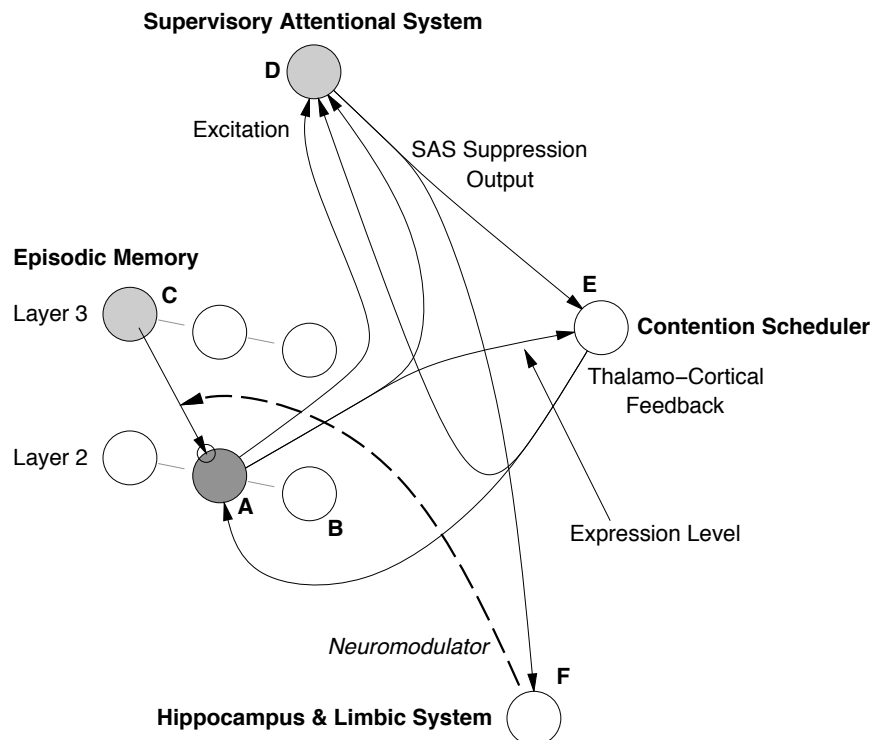


figure 6.10 The process of suppressing an inappropriate node by the Supervisory Attentional System. Node **A** and **B** in episodic memory compete for expression through the contention scheduler marked **E** (for clarity Node **B**'s **Expression Level** and **Scheduler Feedback** signals are not shown). In the diagram Node **A** has become inappropriately selected for expression by the contention scheduler, this is detected by the SAS (Node **D**) as a difference between the **Excitation** (the intention of the entire system to express this node) and **Scheduler Feedback** (thalamo-cortical feedback, indicating the node has become expressed) signals. A warning is generated by the SAS Monitor of Node **D** causing the SAS Modulator of Node **D** to attempt to suppress the expression of Node **A** via the contention scheduler. The SAS Modulator continues to attempt to suppress Node **A**'s expression until Node **A** stops attempting to become expressed (via the **Expression Level** signal). Activity by the SAS also causes the hippocampus node for Node **A** to become active (Node **F**; the link from Node **F** to Node **A**, which when active signals Node **A** is being suppressed by the SAS, is not shown), causing a diffuse neuromodulatory signal to be propagated through episodic (and working) memory. Inhibitory links from appropriately expressed nodes (Node **C**) to suppressed nodes (the link from Node **C** to **A**) are reinforced, so that over time reduced, or zero attentional effort will be required for Node **B** to be appropriately expressed and the correct behaviour to occur automatically

When the affective system becomes active due to raised attentional activity, a modulatory chemical signal is propagated through the entire neural network, reinforcing inhibitory links from active nodes in memory systems that are not suppressed by the SAS to active nodes which are suppressed by the SAS (see *figure 6.10*). Over time, this results in a reinforcement of the association between the attended to episode and expressed action so that, in future, reduced, or zero attentional effort will be required for the appropriate action to occur automatically.

6.2.7. Visualising the Functional Components

Following on from the previous sections, *figure 6.11* illustrates the neural composition of the functional ‘building block’ described in *Chapter 5, figure 5.11*. Neurons and connections shown in *figure 6.11* were taken from SNNS graphical displays of the individual components and include all neurons, but do not necessarily show all the connections.

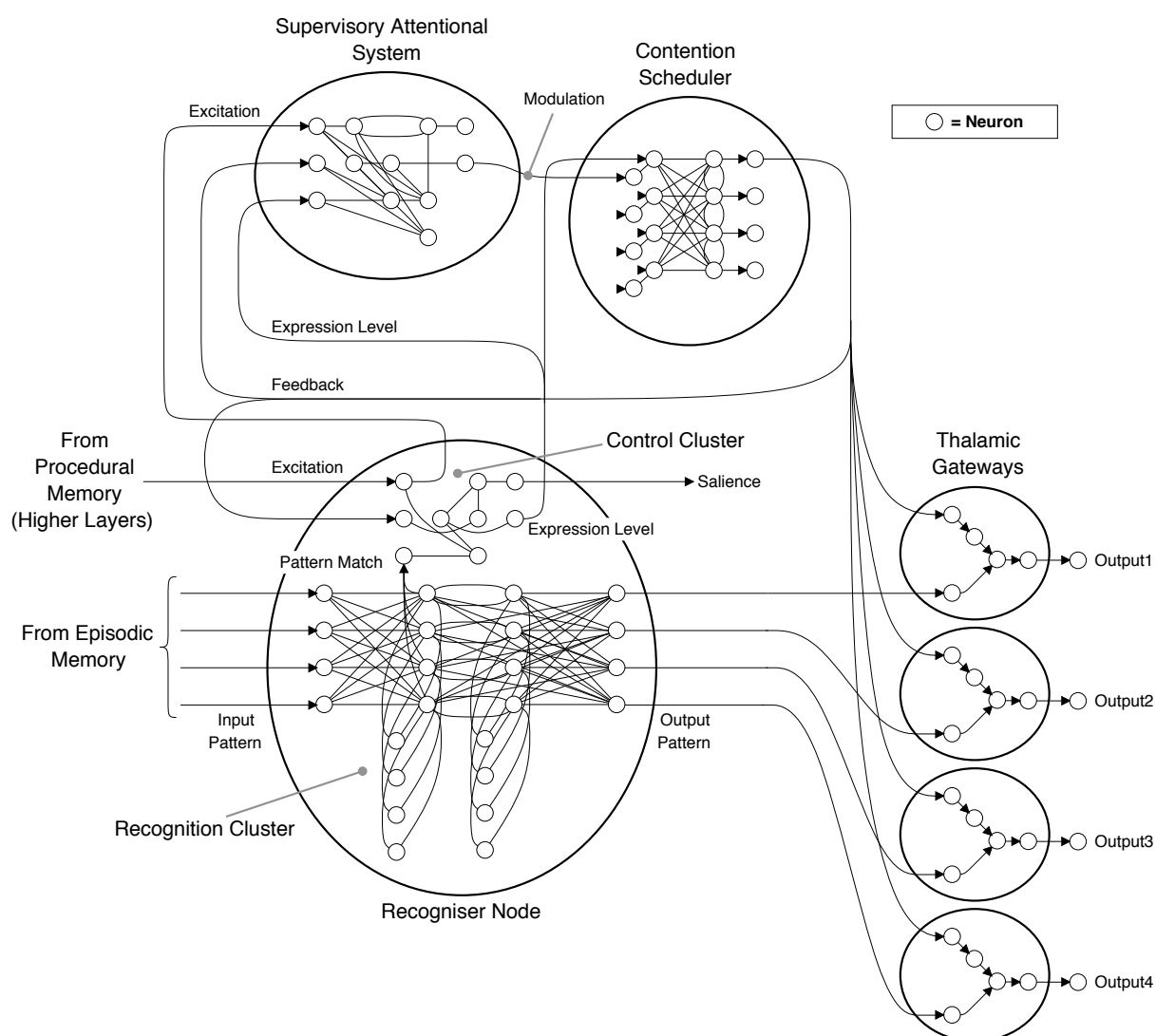


figure 6.11 The same recognition cluster as illustrated in *figure 6.2*, shown at neuron level as part of procedural memory, connected together with its control cluster (to form a recogniser node), contention scheduler, SAS Monitor/Modulator node in the Supervisory Attentional System and associated thalamic gates

6.3. *Controller Training and Operation*

Within the Attentional Architecture as implemented, there are two main types of network.

- **Type 1**, including the SAS Monitors, SAS Modulators, recogniser node control networks, thalamus gates and large recurrent networks such as the contention schedulers are hand built and although they have pattern files to verify input to output mappings are not trained.
- **Type 2** networks such as the sequencing networks in working memory are based on Jordan networks (feeding the output of the network back into itself to move to the next state) whilst type 2 networks in procedural memory are based on Elman networks (using hidden layers to record state). Both were trained in SNNS using a standard back propagation function (JE_BP) for partially recurrent networks. Episodic memory within the implementation is associative and consists of feed-forward networks trained using standard back-propagation tools within SNNS.

All the neural networks that make up the model have associated pattern files, specifying an input and a desired output. For networks in memory subsystems the pattern files were also generated in two ways:

- For episodic memory, pattern files were generated by simple procedural logic, specifying the inputs and generating the pattern file directly
- Working memory and procedural memory pattern files were generated from the simulation. A simulated robot was created and a partial architecture created. Using the simulation the machine was then randomly placed (using the `util.Random` Java class as a random number generator) and moved (again at random) in a specific direction whilst the output of the machine and the partial architecture was captured by procedural logic which then generated the corresponding output and pattern files.

All the pattern files associated with non memory subsystems were generated using procedural logic. Training in SNNS using the pattern files for any network stopped when the output average error value shown by the SNNS Network Analyser tool reached a value of less than 0.1 (10%). Indicating a small deviation from the network generated output to the pattern file output (teaching output).

6.3.1. *Network Evaluation*

The simulation can be made to honour the three modes of network evaluation used by SNNS (specified in the cluster files): synchronous, sequential asynchronous and random asynchronous. In the current simulation, all the neurons are evaluated by a random asynchronous algorithm as the choice of evaluation function can have a significant effect on the dynamics of the networks. Under certain network configurations either synchronous and sequential asynchronous evaluation can lead to limit-cycles [Zurada, 1992, p. 253] in the sequence of network states that the controller exhibits, i.e. it may settle down into a periodic sequence. Evaluating the network using a random asynchronous update

function, assures a 'chaotic' sequence of network states. When a pattern of behaviour appears independent of the precise sequence of network states, as is the case in the results reported in this thesis, there can be considerable confidence that it is robust and reproducible.

Initially, the control network is in an undefined state (recogniser nodes, contention schedulers and SAS nodes have an internal state which may be ill-defined when the networks are first instantiated as the multiple threads do not guarantee the order of evaluation). For this reason 'state-full' clusters have a reset input which can be used to set the cluster to a known state to facilitate controlled experiments. Similarly, the systems that interface to, but are not part of the controller per se (individual sensors and effectors, etc.) have to be enabled for the model to operate. (Note: as implied in the preceding paragraph, starting the network in a known state does not mean that a given sequence of network states follows).

6.3.2. Threading

A thread within Java is classed as 'a thread of execution in a program', with the Java Virtual Machine allowing an application to have multiple threads of execution running concurrently. There are limitations to the threading model, e.g. if one thread needs to access data in another, data has to be synchronised (ensuring that the data is not changing as it is being read) slowing down the system, however for large applications the advantages of being able to fully utilise machines with multiple central processing units across symmetric multiprocessing (SMP) or non-uniform memory access (NUMA) machines far outweigh the disadvantages.

The graphical interface (see *Chapter 7, section 7.1*) is always run in its own thread, as is each simulated machine operating in the environment. Each machine has its own instance of the control architecture which can have several tens of its own threads. To choose whether a cluster will run in its own thread or be part of a larger group the 'thread' attribute can be used with any of the elements in the XML file that specifies the control network (see *listing 6.14*); setting this to true will create a new thread and name it the same as the element (to assist performance monitoring and debugging). Each thread has an associated network (see *section 6.1.3*) to which neural clusters will be assigned until another thread is created. All threads run at the same priority. Within the implementation described in *Chapter 7*, eleven threads make up the control architecture for one simulated machine.

The SheepWorld3 simulation system has been tested with the Attentional Architecture on machines with one, two and eight CPU's at various levels of threading, with different versions of the Java Virtual Machines (1.4 & 1.5) on different machine architectures (Apple Mac OS X 10.3 & 10.4, IBM AIX 5.2 & 5.3, Sun Solaris 2.6, 2.8 & 10, GNU/Linux (2.2, 2.4 & 2.6 kernels) and Windows 2000/XP). It would not appear to make any difference to the observable behaviour of the machine as to the architecture, whether one or twenty threads are used for the simulation nor the number of CPU's other than in the speed of the simulation and the number of evaluations that can be done per second.

Setting the number of threads to around two to four times the number of CPU's in the system gives very acceptable performance, i.e. using a single 2.4GHz Intel Pentium 4 CPU, with 2GB of memory (of which the simulation uses about 256MB) the simulated machine completes a single run of the experiment (see *Chapter 7, section 7.3*) by moving from the start point to home in under ten seconds.

```
<!-- Three thalamic gates running in three threads -->
<nuronet name="gate_1" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="true"/>
<nuronet name="gate_2" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="true"/>
<nuronet name="gate_3" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="true"/>

<!-- Three thalamic gates running in one thread -->
<nuronet name="gate_1" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="true"/>
<nuronet name="gate_2" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="false"/>
<nuronet name="gate_3" cluster="thalamic_gateway.net" type="SNNS" dupl="1" thread="false"/>
```

listing 6.14 An XML listing showing how to control threading in the creation of the control architecture, the first fragment creates three separate threads in which one cluster is evaluated in three parallel streams. The second fragment creates one thread in which three clusters are evaluated.

6.4. Summary

This chapter has described the detailed implementation of the Attentional Architecture from the ideas developed in *Chapter 4* and *Chapter 5*. It has shown how the elements of the model are specified and managed using XML and how this XML specification is used to construct the ‘runtime’ computational system (the Attentional Architecture) and concludes with a description of how the functional neural clusters were trained together with the ‘runtime’ characteristics of the simulated control architecture.

Chapter 7

7. Behavioural Properties

This chapter examines the behavioural properties of a simulated robot controlled by an instance of the architecture developed in previous chapters. It begins with an account of the simulated environment, the machine that functions within it and how the activity in the control network can be recorded. It then describes a specific instance of a controller constructed in accordance with the specified architecture. Thereafter it presents a series of studies using the simulation. The first studies show the ‘normal’ functioning of the machine, the major feature of which is the way in which the attentional mechanism responds to distraction and progressively learns to avoid it. Having established normal behaviour, there ensues a number of lesion studies which examine the effects of a variety of lesions at the architectural level, that is at the level at which the attention control system interacts with the other subsystems of reactive control system. Discussion of these results follows in *Chapter 8*.

7.1. The Simulated Environment and Robot

The default simulation environment is shown in figure 7.1; it is essentially a boundless field into which multiple objects and simulated robots can be placed. All objects within the environment have attributes such as colour, a chemical odour and height which can be manipulated directly through the user interface or programmatically.

The simulated environment can be interacted with directly by creating objects (clicking into empty space when in create mode), objects and machines can be picked up, moved or rotated at will

To speed up or slow down the simulation, use the slider to alter the time increments for the global clock, note that this does not alter the time for the controlling network, it is always run as fast as possible

Start and stop the global time and any threads that the controlling network and robots are using

Scroll or zoom into or out of the simulated environment

Create a new simulated robot

Messages are generated by almost all of the simulation and are filtered and logged by SheepWorld3, important messages are displayed here, a separate log window can display all the messages (i.e. network creation and debug information)

The Reset button is connected to a sensor on every robot in the simulation, this can then be connected to any necessary reset logic (i.e. contention schedulers) before the start of an experiment

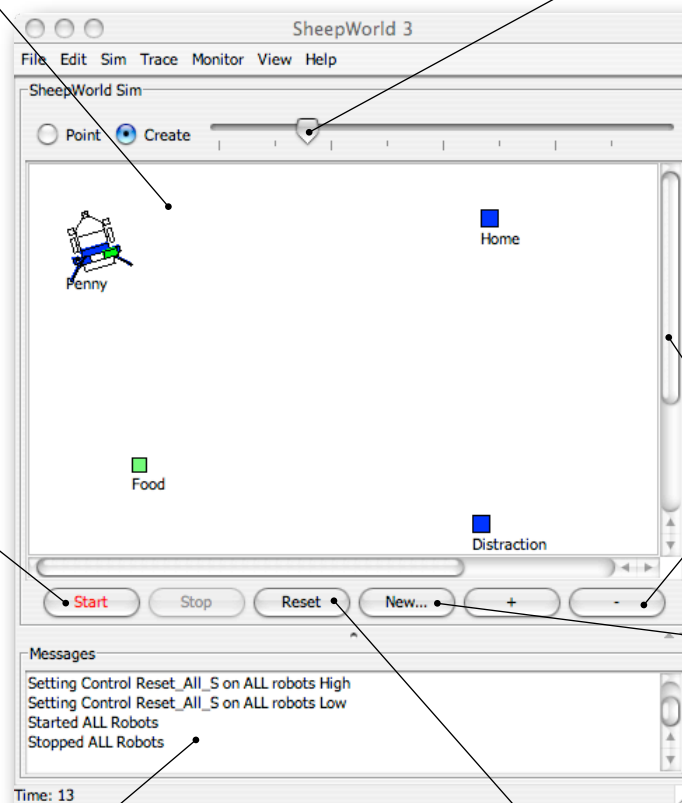


figure 7.1 The main canvas of SheepWorld3 (generated by the simulation), showing Penny the simulated robot together with a home location and a food object

7.1.1. Robot Sensors and Actuators

The simulated robot (named Penny) has two, forward facing sonar sensors and eight olfactory sensors that are placed to allow it to sense the presence of obstacles and objects of interest such as food, home, and other robots if present. Sensor or actuator data is interfaced to the Attentional Architecture using specialised input and output neurons (these all have identity transfer functions).

The simulation implements sensors and actuators using the standard techniques and algorithms prescribed in [Dudek and Jenkin, 2000]:

- Olfactory sensors [Dudek & Jenkin, 2000, p. 80] able to detect the concentration of simulated chemical odours (which can be specific to each object type, e.g. food, home, another robot). Concentrations are assumed to follow an inverse square law based on radial distance from the source object. Individual olfactory sensors are 'tuned' to a specific odour. Sets of four sensors placed at each corner of the simulated robot provide data on the resulting chemical gradient, which are interpreted by a trained neural network to generate a vector in the direction of the source. Two sets of four sensors (food and home) are used in the experiments described in this chapter
- Sonar or 'time of flight' sensors [Dudek & Jenkin, pp. 59-64] can be used to determine distances to objects. These active sensors fire an ultrasonic pulse (over a 60° arc) and measure the time taken for an echo to be received. The elapsed time is presumed to be directly proportional to the distance from the object the ultrasonic pulse was reflected from. Multiple reflections from objects in the environment (generating spurious or ghost objects) are simulated but variability of an objects surface reflection or absorption of the pulse is not. Two sonar sensors are mounted on a sensor 'head', looking forward, that can be rotated 45° left or right to sweep or 'saccade' across the field of view as shown in *figure 7.2*
- Each machine has a simple gripper, that can open and close around objects of interest in the environment. A small micro-switch situated between the two fingers can sense when the gripper is placed around something that it can pick up and is also activated when the gripper is closed around an object (i.e. the object is held against it)
- A differential drive [Dudek & Jenkin, 2000, pp. 18-22] is used to move the robot around the simulated environment with each motor having a speed and a forward or reverse setting. The motors are positioned on each side of the robot so that it can rotate around its own central axis in a clockwise or anticlockwise fashion. Other sensors and actuators, such as a simple colour vision system, sound generation and sensing, infrared sensors and micro-switch based whiskers are available but not used in these studies.

Although most of the power in a sonar pulse is near the centre of the beam, significant side 'lobes' exist. The real sonar devices, spread the pulse over a 60° arc that can be seen in the 'cone' of trace rays from each sensor

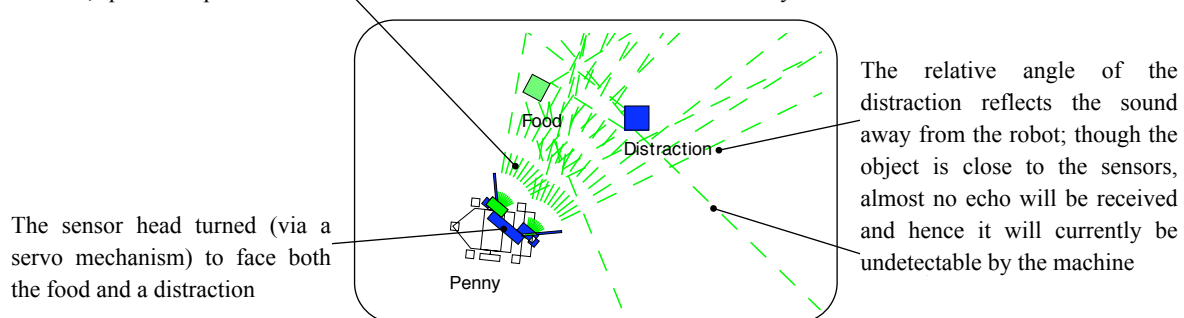
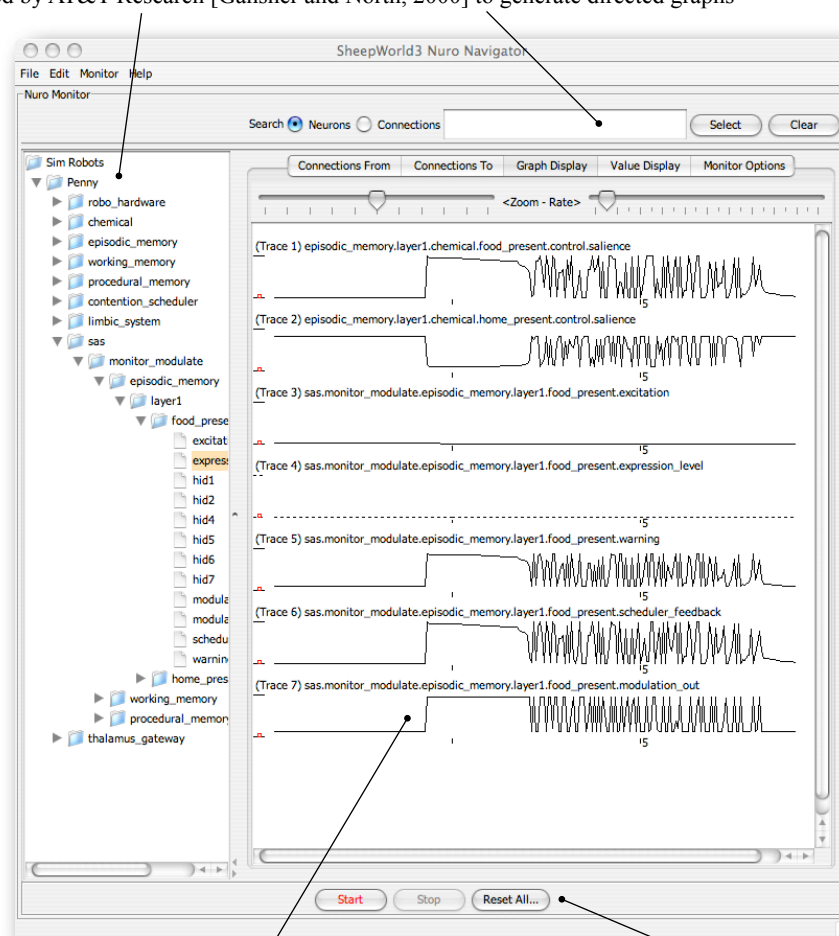


figure 7.2 The robot Penny, depicting the traces (dashed lines) from the two sonar sensors

7.1.2. Observing the Controlling Network

The simulation environment also allows monitoring of the activity of the controlling network. *Figure 7.3* shows how the architecture for each controller and robot are displayed using the object naming convention previously described. Activity in collections, networks and individual neurons can be displayed or traced and connections can be viewed, altered in strength or removed (lesioned).

The hierarchical view of the architecture is the main interface to the instantiated model, grouped by robot; collections, networks, clusters and individual neurons can be selected for display, monitoring or modification using this mechanism. In large architectures neurons and connections can be selected using the text search box (shown). Though the ‘structure’ of the architecture is not directly displayed by SheepWorld3, all or parts of the model can be output directly in ‘dot’, a language created by AT&T Research [Gansner and North, 2000] to generate directed graphs



Once a set of neurons has been selected, it can be displayed by the Navigator panel in a number of ways. Connections From and To allow for the viewing and modification of connections from/to neurons. The ‘Graph Display’ can display the output or (summed) input of a neuron as a trace and can be used as in a similar way as a logic analyser to correlate and capture events happening within the model in realtime. An adjustable zoom and sample rate enables the recording of rapid transients. In order to visualise patterns across hundreds of neurons the ‘Value Display’ panel can be used to show the input or output value of sets of neuron as coloured pixels, ranging from black (0) to green (1) through to red (2+)

Start, stop and reset buttons control the Navigator and any associated displays. Data captured by the Graph display can be logged (together with a global time) to a CVS file for analysis by external software

figure 7.3 Monitoring controller activity in context is done using the Navigator interface

7.2. An Instance of the Attentional Architecture

An instance of the control architecture has been created which has the capacity to accomplish a goal of foraging for food (*figure 7.4*).

It has six hundred and eight individual neurons in sixty three neural clusters with one thousand and seventy one connections.

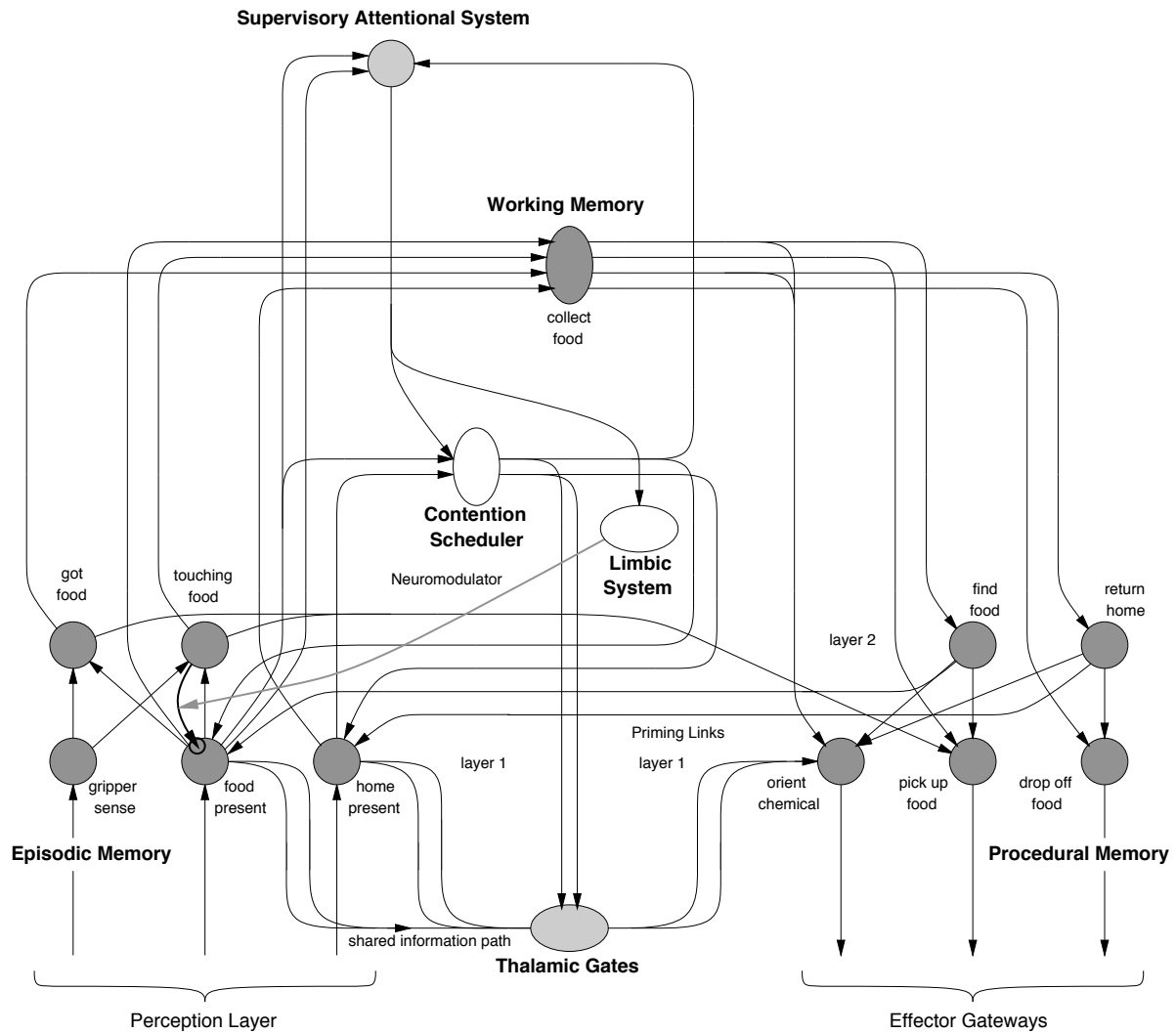


figure 7.4 The instance of the Attentional Architecture used in the studies showing the nodes and their connections. A recogniser node in working memory 'collect food' sequences the pattern of procedural and episodic memory excitation for the foraging behaviour studied in this chapter. Priming links from procedural memory to episodic memory are shown (equivalent priming links from working memory to episodic memory are not). In this model four parts of the contention scheduler are operating in parallel, the first part (shown) is used by 'food present' and 'home present' to access the shared information path to 'orient chemical', the second (not shown) protects the 'collect food' recogniser in working memory and two others (not shown) protect the actuators and grippers in procedural memory. The inhibitory link from 'touching food' to 'food present' sensitised to neuromodulators released by the limbic system is shown with a circle around its arrow tip

To achieve its assigned goal, the robot must undertake a series of sub-tasks: make its way to the food, collect it and then take it to a specified location (home). In the absence of any other stimuli, the normal sequence of events proceeds as follows:

- Assuming both food and home are in the environment and sensed via perceptual input, the behaviour 'collect food' is initiated by working memory 'exciting' the procedural memory node 'find food' and, indirectly, the episodic memory node 'food present'. Once active this episodic memory node has to compete with 'home present' for reactive access to a shared information path through the contention scheduler to procedural memory.
- The behaviour node 'orient chemical' causes the robot to move towards a chemical signature, in this case, food.
- Once the machine's gripper senses the food, the episodic memory 'touching food' becomes salient, triggering the next sequence in the plan, 'pick up food'. 'pick up food' halts the machine and attempts to close the grippers, if this is successful 'got food' becomes salient.
- The activation of 'got food' induces working memory to activate 'return home' in procedural memory together with 'home_present' in episodic memory which together with 'orient chemical' orientates and moves towards home (competing with 'food present' for exclusive access to the shared information path to procedural memory).
- As the salience of 'home present' reaches a threshold (proportional to distance from home) the next sequence in the working memory plan activates 'drop off food' which again halts the machine and the foraging cycle is restarted.

7.3. Normal Behaviour and Learning

There are many clinical 'attention' tests, which aim to evaluate various aspects of attention in a patient with lesions [van Zomeren and Spikman, 2003, p. 76], they are grouped into one of the following types:

- **operational tests**; where the time to respond to a stimulus is measured. These examine perceptual attention and are stimulus driven in a highly structured experiment (e.g. responding to patterns). The Stroop test (see *Chapter 3*) is an example of an operational test
- **tactical tests**; under some moderate time constraints, these tests aim to assess the monitoring and modulation aspects of executive attention. The test is usually memory driven and given in a partially structured experiment (e.g. ignoring distractions)
- **strategic tests**; these tests are set with low time thresholds in an unstructured environment where the subject has to find their own approach or optimum solution to an unfamiliar task (e.g. sorting cards in the Wisconsin Card Sorting Test [van Zomeren and Spikman, 2003, p. 81]) to investigate strategy generation and overall supervisory control

When testing behaviour over time, both tactical and strategic tests are appropriate. As the SAS Generator has not been implemented a ‘strategic’ test of the system is outside the scope of this thesis. Hence a ‘tactical’ experiment is used.

For the purposes of the work here ‘foraging’ has been chosen as a tactical test. Many insects, birds and mammals exhibit ‘foraging’ which is an observable goal based behaviour usually consisting of finding, picking up and then hoarding (transporting to a specific location) food or nesting material.

As a tactical test foraging provides a readily observable high-level behaviour which is achieved through the exhibition of a sequence of lower-level tasks, where the task sequence may be performed automatically (that is without recourse to attentional control) in the absence of distraction but may be modulated by attentional effort in the presence of unfamiliar sources of distraction.

The test described in this thesis involves adding a significant distraction (additional food) whilst the machine is engaged in taking home food it already holds.

Using foraging as an experimental tool has a long history in based behaviour based robotics. The subsumption based robot Herbert [Brooks et al., 1988] was designed to wander around University offices and ‘steal’ empty soda-cans, taking them to a pre-programmed central location. Mataric [1996] used foraging on simulated and physical machines as a demonstration as to how basic (basis) behaviours could be combined into higher level individual (and group) behaviours. More recent experiments by Kasderidis and Taylor [2003] involved using an extended attention based architecture to forage for objects and return these to a specific location (see *Chapter 3, section 3.4.4*).

7.3.1. Normal Behaviour in an Inexperienced Machine

The normal behaviour of an inexperienced machine (newly instantiated by the simulation) is shown in *figure 7.5* and *figure 7.6*. As a wholly inexperienced machine, it has the capacity to forage successfully, but it has not learnt how to do so robustly.

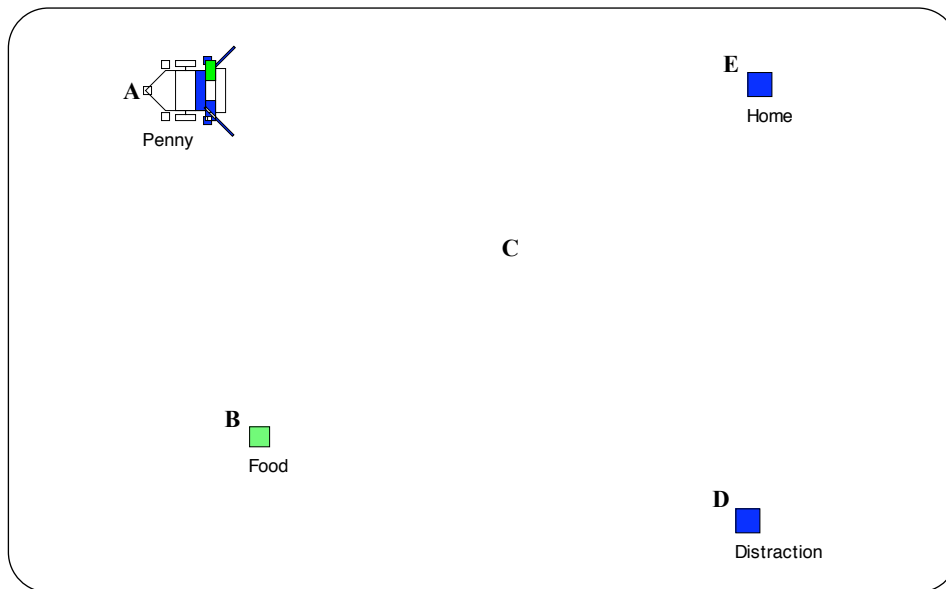


figure 7.5 Initial configuration of the environment for the tactical test for distraction

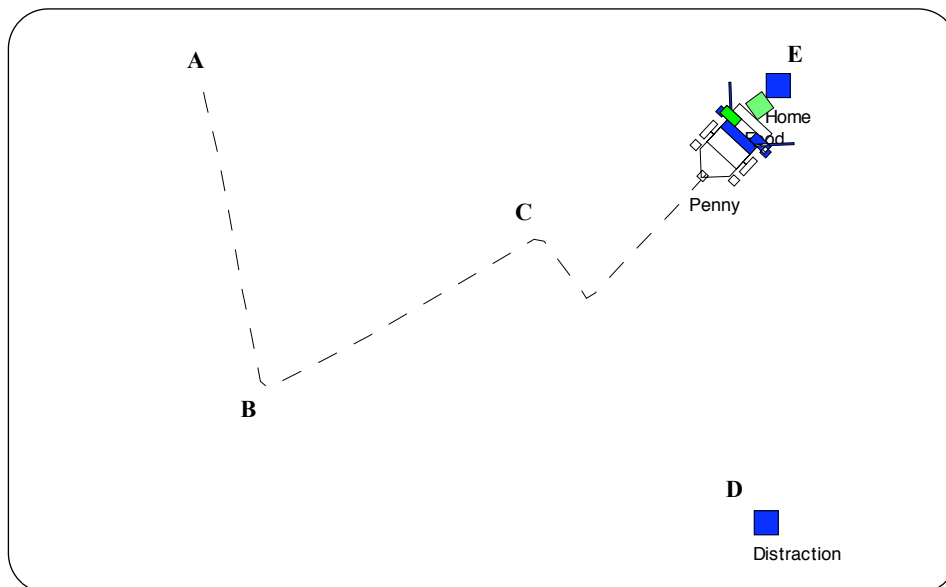


figure 7.6 The final position upon completion of the experiment. The dashed grey line denotes the path penny has taken from point A to point E via points B and C

In the two figures above the robot is learning to deal with an unfamiliar distraction during execution of the plan to forage held in working memory. In *figure 7.6* the foraging machine starts at point A, food is placed at point B and home is at point E. After the robot has collected the food at B additional food (a highly salient distraction) is introduced into its environment at point D when the machine is at point C.

The effect of the distraction is evident in the path of the machine. When we observe the activity of selected neurons within the control network, a more subtle picture emerges. The traces of *figure 7.7* show the activity of the ‘food present’ (trace 1) and ‘home present’ (trace 2) episodic memory nodes as they compete for expression as well as the inputs and outputs of the SAS Monitor and Modulator connected to ‘food present’ (traces 3-7).

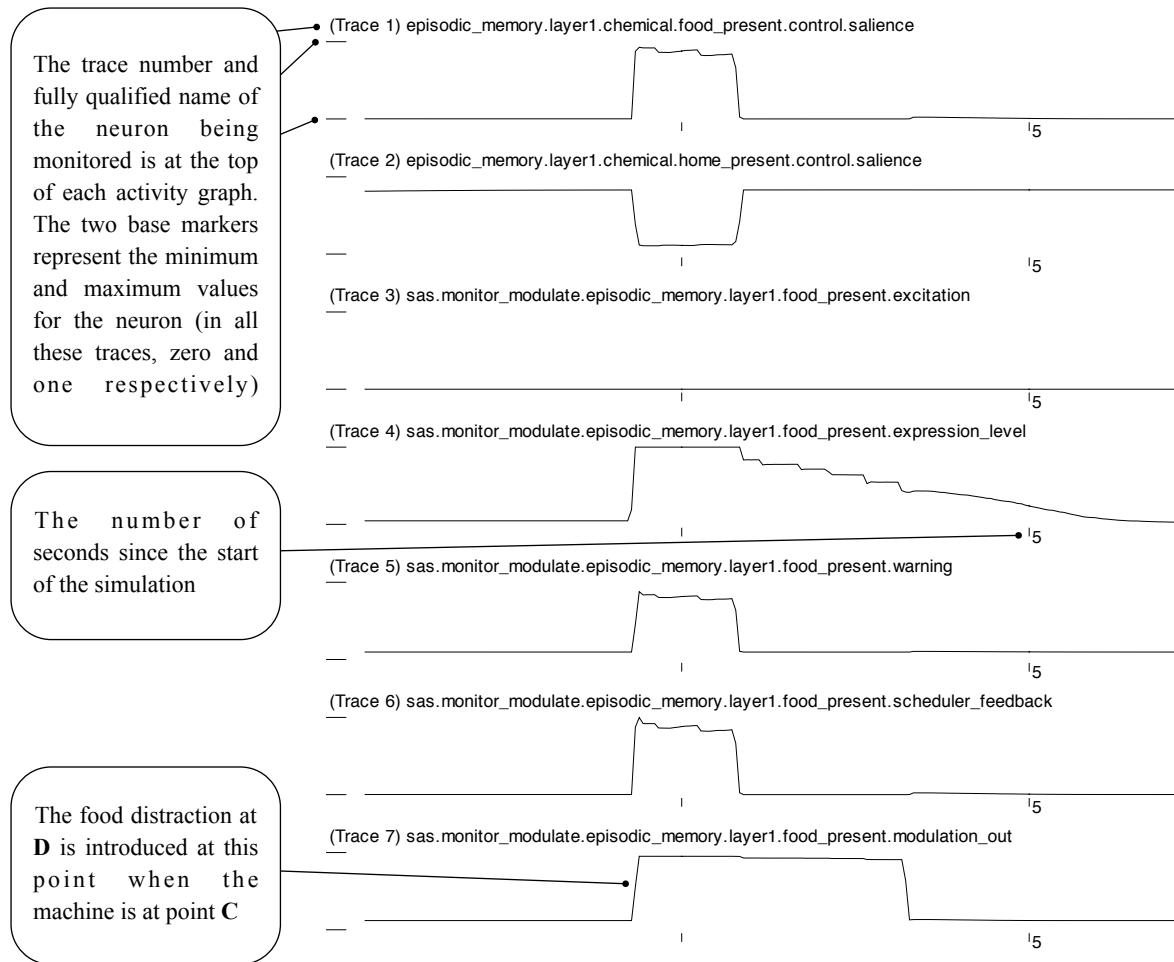


figure 7.7 Traces from the simulation showing the activity of seven neurons from the model as the simulated machine moves from point **B** to point **E**. The distraction is introduced at **D** whilst the machine is at point **C**.

The traces in figure 7.7 were initiated a short time after the machine has collected food at point **B** but before it has reached **C** where the distraction is introduced; consequently the traces start with the intended expression of ‘home present’ (trace 2) whilst ‘food present’ (trace 1) is inactive. The traces record the activity of selected neurons from the model as the simulated machine moves towards home (point **E**).

The distraction, **D**, is introduced at whilst the machine is at point **C**. As soon as the distraction is introduced, the salience of the episodic memory recogniser node ‘food present’ increases (rising edge of trace 1). This level of salience is sufficiently high for it to be expressed via the contention scheduler. This expressed behaviour is visible to the SAS (trace 4) and is the basis of reinforcing feedback to the active node (trace 6). In consequence, the (relative) salience of ‘home present’ diminishes (trace 2). At this stage of the foraging behaviour there is no intention to respond to the presence of food (trace 3). The level of this intention is primarily driven by the state of working memory but includes an inhibitory input from lateral connections in episodic memory, principally from ‘touching food’, sensitised to the neuromodulator released by the limbic system. The SAS Monitor responds to the

misalignment of the intended response to the presence of food (trace 3) and the expressed response (trace 4), producing an arousal or warning signal (rising edge of trace 5) which can be seen by the SAS Modulator. The response of the SAS Modulator is to attempt to suppress the node 'food present' by means of an inhibitory signal to the contention scheduler (trace 7). This attenuation is eventually successful (falling edge of trace 6 from the contention scheduler, and consequently of trace 1). This allows 'home present' to regain expression (recovery of trace 2).

Meanwhile, the modulation signal to the contention scheduler (trace 7) is also used as a basis for a 'learning' signal to the limbic system (hippocampus), which in turn induces the release of a diffuse neuromodulatory chemical that increases the weighting of lateral inhibitory connections from other active (and not suppressed by the SAS) nodes in episodic memory to nodes which are suppressed by the SAS but still active (in this instance, the connection from 'touching food' to 'food present'). Accordingly the strength of the inhibitory connection from the intended node of 'touching food' to the active but unintended node 'food present' begins to increase. Though the SAS has attempted to suppress the expression of 'food present' almost immediately (as can be seen from the trace in *figure 7.7*, the machine still begins to move towards the distraction); 'food present' continues to attempt to express itself (continued activity from trace 4), requiring sustained attention by the SAS (plateau of trace 7). Eventually, the neuromodulator-mediated attentionally derived learning signal increases the salience of the inhibitory link between 'touching food' and 'food present' to the point where it suppresses its activity and 'home present' regains expression (recovery edge of trace 2). SAS modulation continues however until 'food present' stops requesting expression (the gradually falling activity of trace 4) and the SAS modulation can cease (falling edge of trace 7). At this stage, the robot has learnt to avoid the level of the distraction encountered and no longer needs attentional resources to act appropriately in the presence of a similar level of distraction

7.3.2. Behaviour in an Experienced Machine

In a second experiment (a continuation of the first experiment, after the machine has arrived home), on this partially experienced machine the salience of the distraction is increased by bringing it closer (about half the original distance from when first encountering the distraction). The resulting controller activity is shown in *figure 7.8* which shows the effect is comparable to the first experiment with the machine learning to overcome the more salient distraction

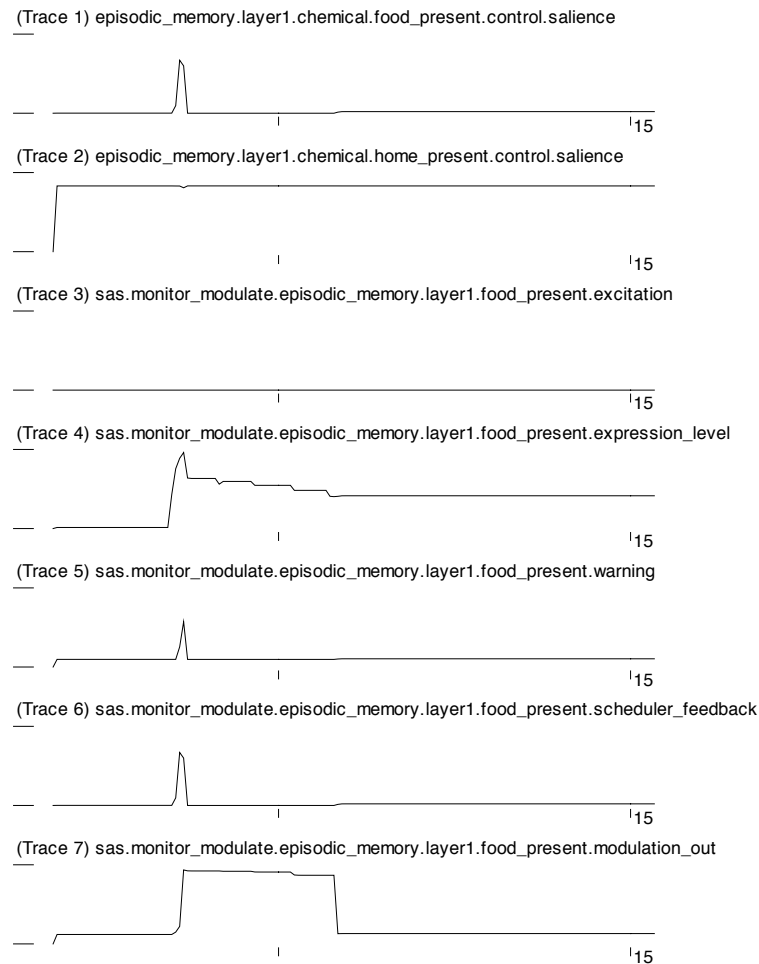


figure 7.8 In a second experiment, these traces illustrate the introduction of a more potent distraction to the machine whilst it is located at home. The spike on traces 1, 5 and 6 correspond with the introduction of the distraction

In a third run of the experiment the distraction is brought immediately adjacent to the machine (*figure 7.9*), and the resulting traces, again following the same pattern, are given in *figure 7.10*.

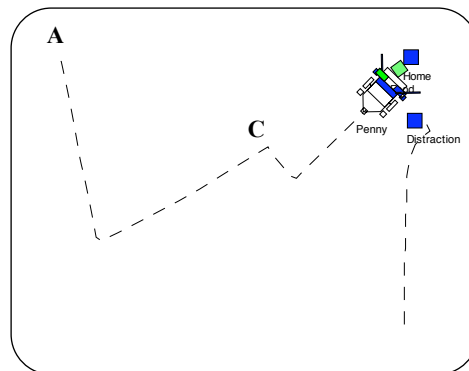


figure 7.9 The distraction is given maximum salience by placing it adjacent to the machine

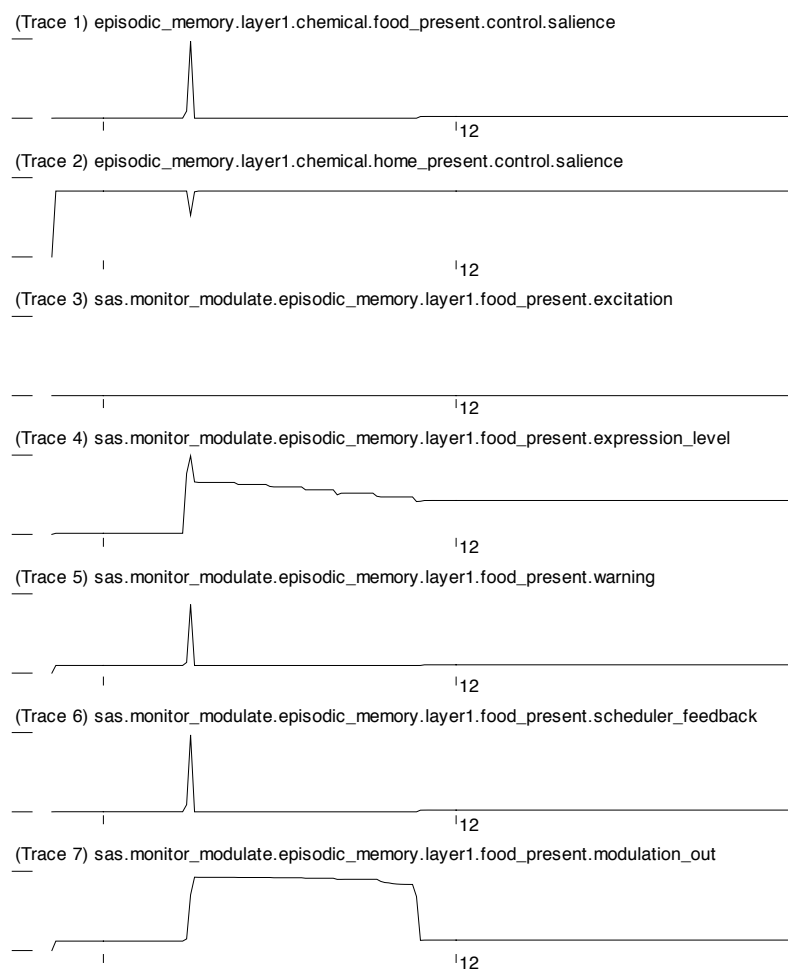


figure 7.10 Traces from the third run of the simulation for a partially experienced machine. The spike on trace 5 shows when the distraction is introduced adjacent to the machine

In the final run of the same experiment (the network file was saved at the end of the first experiment and reused) shown in the traces of *figure 7.11*, the robot demonstrates that it has learnt to ignore even the most salient distraction, and requires no intervention of the SAS to accomplish its goal

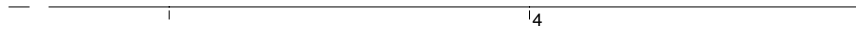
(Trace 1) episodic_memory.layer1.chemical.food_present.control.salience



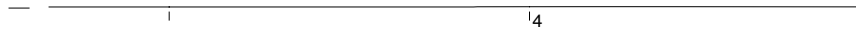
(Trace 2) episodic_memory.layer1.chemical.home_present.control.salience



(Trace 3) sas.monitor_modulate.episodic_memory.layer1.food_present.excitation



(Trace 4) sas.monitor_modulate.episodic_memory.layer1.food_present.expression_level



(Trace 5) sas.monitor_modulate.episodic_memory.layer1.food_present.warning



(Trace 6) sas.monitor_modulate.episodic_memory.layer1.food_present.scheduler_feedback



(Trace 7) sas.monitor_modulate.episodic_memory.layer1.food_present.modulation_out



figure 7.11 A final run of the experiment using an ‘experienced’ controller taken from the end of the first experiment shown in figure 7.6. When the distraction is again introduced with the machine at point C (3.8 seconds into the trace), no intervention from the SAS is required for the machine to complete its goal and avoid this particular distraction in a wholly unattended way.

Figure 7.12 serves to illustrate that the inexperienced machine’s behaviour is independent of different starting configurations.

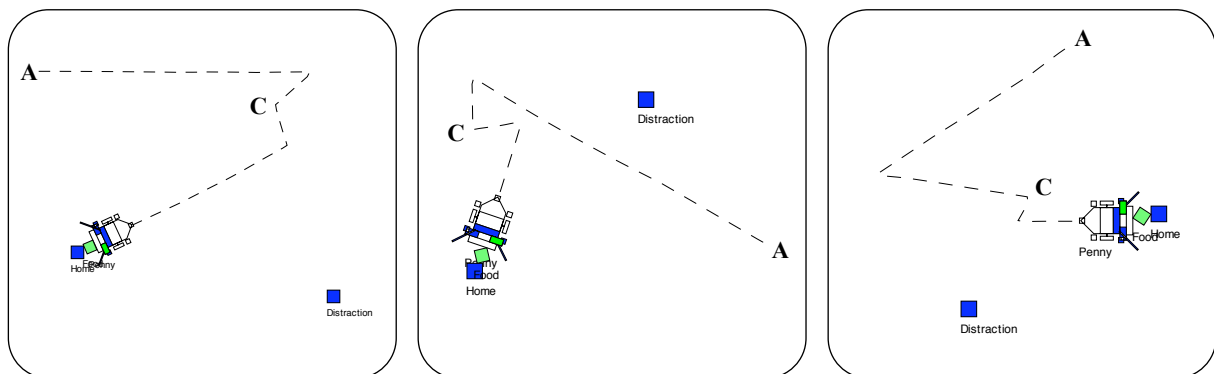


figure 7.12 Performance of a wholly inexperienced machine in different configurations of the test environment

7.4. *Lesion Studies in the Attentional Architecture*

An important tool in cognitive neuropsychology for studying attention in humans is the use of lesion studies [Parkin, 1996]; by recording, analysing and interpreting behaviour when the brain is damaged theories can be developed as to what components are operating and how they relate to each other. However as most lesions are investigated after the lesion (or ‘insult’) has occurred almost all these studies have to be made without direct reference to the normal behaviour of the subject concerned. Though the location and extent of these lesions may be inferred using one of a number of scanning technologies, the lesion may only be properly investigated by post mortem examination.

Lesion studies have also been used to interpret the properties of artificial neural systems [Houston and Sumida, 1985; Hinton and Shallice, 1991; Plaut and Shallice, 1993; Hinton et al., 1993] and to validate artificial neural networks by looking for correspondence with human lesions [Taylor, 2003; Taylor and Fragopanagos, 2003; Fragopanagos and Taylor, 2004; Taylor and Fragopanagos, 2004].

As previously mentioned the simulation interface enables modifications to the controlling network to be made in real time, and the effects observed and captured in traces. The effects on normal behaviour of the lesions allow the properties of the model to be explored as it tries to complete its goals, and can suggest similar behavioural pathologies associated with dysfunction of human executive attention.

7.4.1. *Lesioning the Attentional Architecture*

The structural relations between the SAS and other elements of the Attentional Architecture allows for five strategic simulated lesion studies to explore the induced behavioural pathology in the control network. The locations of the experimental lesions to the attentional architecture (A - E) are shown in figure 7.13. The five lesion experiments (A-E respectively) are:

- **Lesion A** involves removing the stimulus path between the SAS Modulator and the limbic system (the hippocampus), effectively preventing attention based learning
- **Lesion B:** removes the modulatory path between the SAS Modulator and the contention scheduler preventing the SAS from attempting to directly attenuate an inappropriately expressed behaviour
- **Lesion C** prevents the SAS Monitor being aware of the salience of recogniser nodes in episodic memory seeking expression via the contention schedulers as a result of excitation from the rest of the network and the currently perceived state of the environment.
- **Lesion D** prevents the SAS Monitor being aware of the level of excitation reactively induced in a behaviour as a result of the current pattern of behaviour in the rest of the system (i.e. excitation from working memory, from episodic memory or priming from procedural memory)
- **Lesion E** removes the contention scheduler input to the SAS Monitor, removing awareness of expressed behaviour

Note that the trivial lesion, which simply removes the effect of the SAS by combining lesions A and B, is not included. This simply reduces the controller to purely reactive and the observed behaviour is that of a robot that is highly distractible.

The Supervisory Attentional System

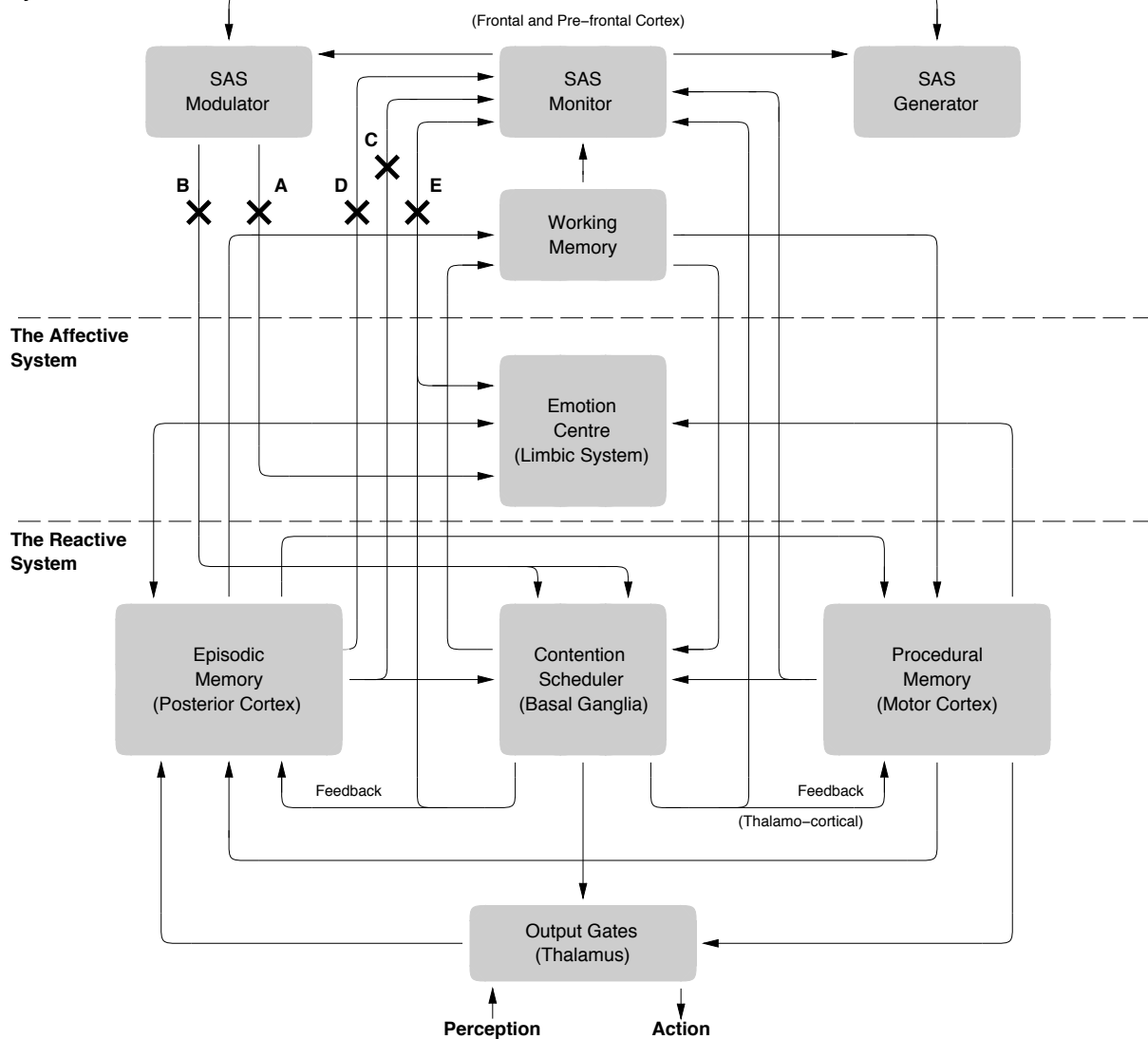


figure 7.13 The five structural lesions (A - E) to the Attentional Architecture. This is an adapted version of figure 4.2 from Chapter 4 and explicitly shows the 'excitation' connection from episodic memory to the SAS Monitor for lesion D

At the implementation level, these pathway lesions are made to connections within each of the 'building block' structures in episodic memory that group nodes with the elements of the SAS and contention scheduler (figure 7.14).

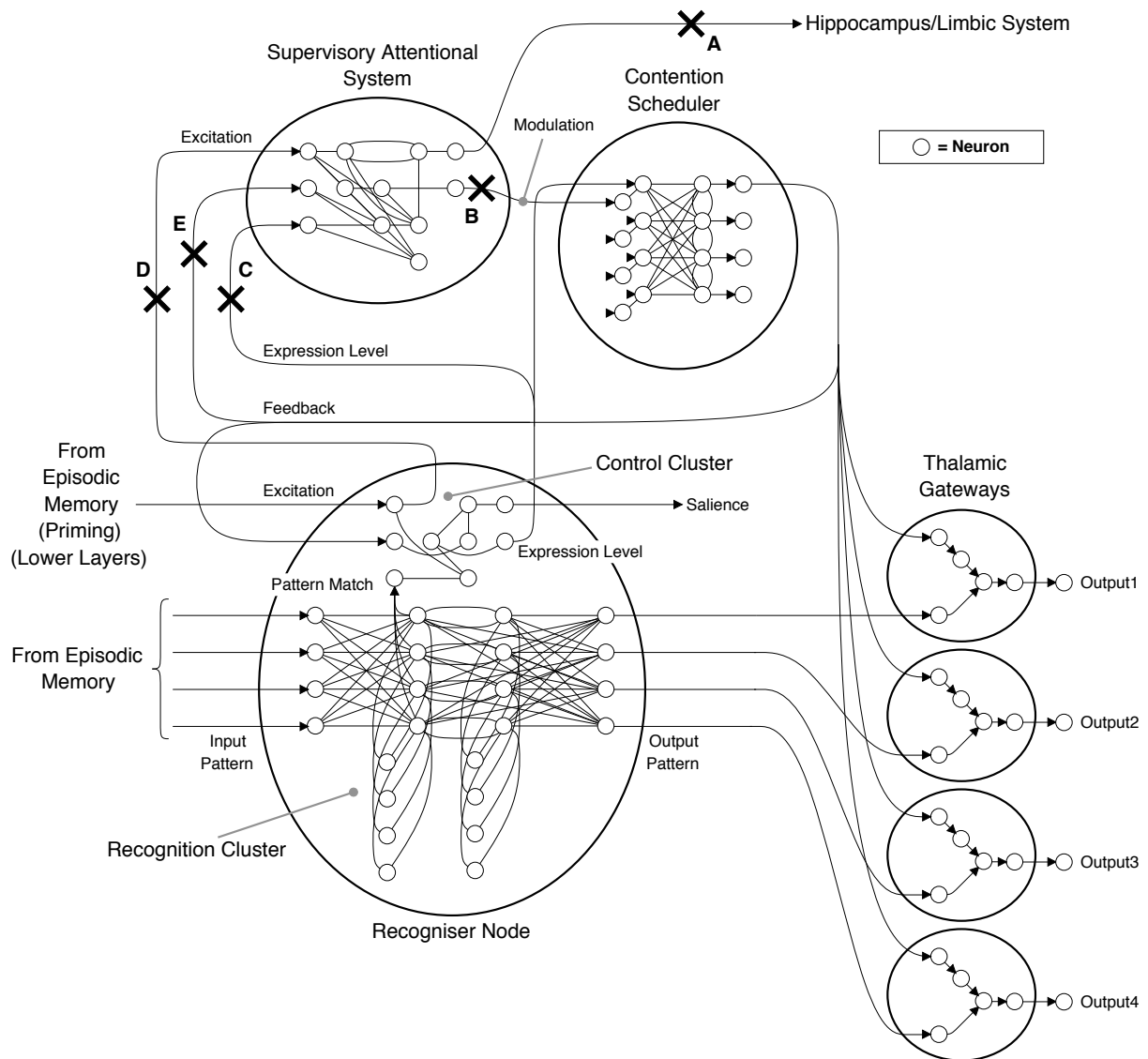


figure 7.14 Lesions at the implementation level, illustrating the building blocks in episodic memory (adapted from Chapter 6, figure 6.11)

7.4.2. Setting up the Simulation for Lesioning

In each simulation run performed for the lesion studies, the machine was allowed to run from point **A** to collect the food at point **B** (figure 7.15) before being halted to allow the controller to be lesioned at point **F**. The simulation was then restarted to complete each experiment.

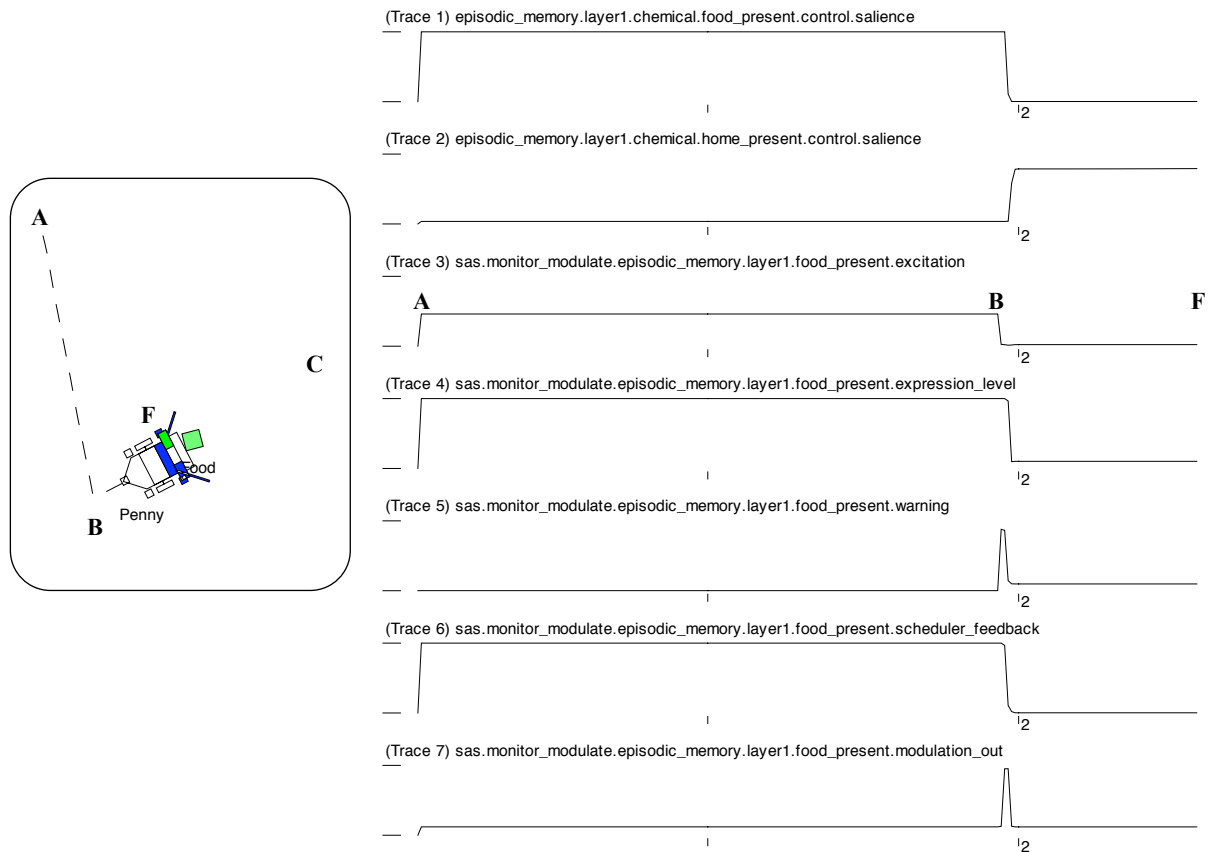


figure 7.15 The preliminary path taken by the inexperienced robot (Penny) and its associated neural traces prior to lesioning the controller.

The simulation starts at point A and the contention scheduler selects behaviour for ‘food present’ in preference to home present (traces 1 and 2 respectively). When food is encountered at point B, it is picked up before the machine starts to move towards home. Note how behavioural persistence in this inexperienced machine during its first run requires momentary executive attention by the SAS to move from one planned state to the next (shown by the spikes in traces 5 and 7). Shortly after it acquires the food but, before it gets to point C where the distraction is introduced, the machine (and the trace) are halted at F so that the controller can be lesioned

7.4.3. Lesion Study A

Lesion study A, consists of lesioning a relatively inexperienced robot by removing the excitatory signal from the SAS Modulator to the hippocampus and the limbic system as shown in figure 7.13 and figure 7.14 marked A. This lesion effectively prevents the machine from capturing (learning) the adaptations that the SAS is making to memory subsystems in dealing with novelty.

7.4.3.1. Lesion Study A: Normal Level of SAS Modulation

If the activity traces of the lesioned machine in figure 7.16 are compared with those of a normal, but similarly inexperienced robot in figure 7.7, it can be seen that outwardly the robot behaves normally (the path taken from point A to C, and traces 1, 2, 5 and 6) until a little while after point C, where it

moves towards the distraction. However because attention based learning via the hippocampus is not taking place, the modulation the SAS has to apply (trace 7) is constant and unlike the normal inexperienced machine it never escapes the distraction, but keeps moving forward. The activity of ‘food present’ (trace 4) is never lowered until the distraction is moved away and hence this machine will never acquire the ability to exhibit the correct behaviour without significant attentional effort.

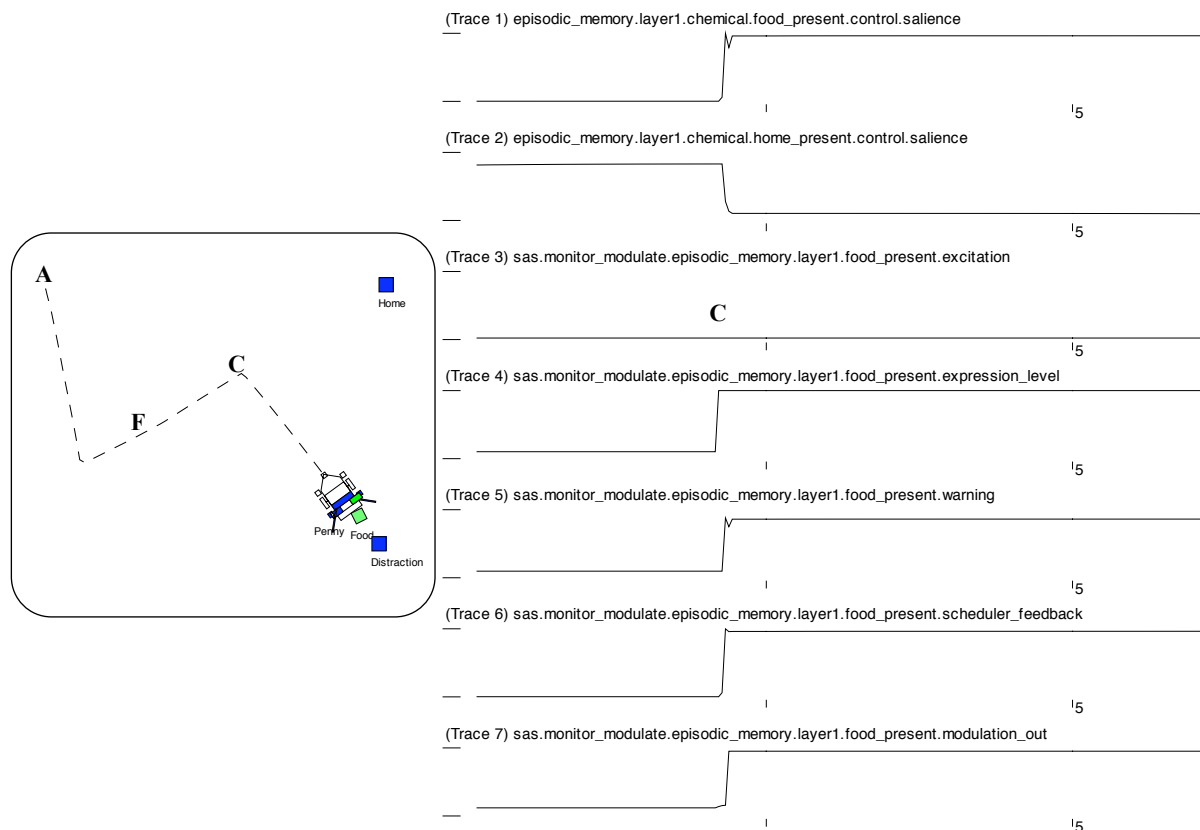


figure 7.16 The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study A; the robot starts from point A, the traces start from point F. A distraction is introduced when the machine is at point C (and shown on trace 3)

7.4.3.2. Lesion Study A: High Level of SAS Modulation

A second set of traces are shown in figure 7.17, these illustrate the effect of doubling the weight of the connection from the SAS Modulator to the contention scheduler (from 1.0, the default to 2.0) as well as introducing lesion A.

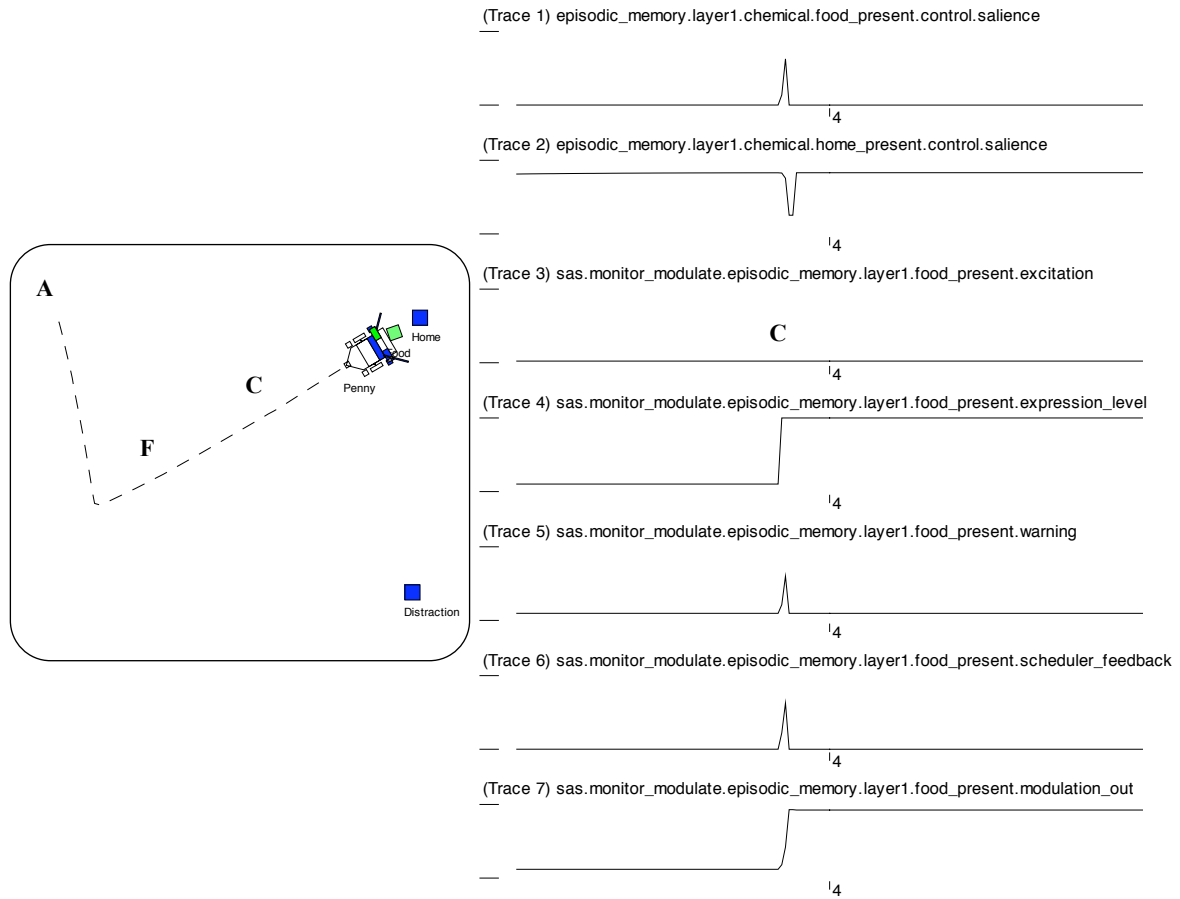


figure 7.17 A second run of lesion A, with the weighting between the SAS Modulator and the contention scheduler doubled.

It can be seen from the traces in *figure 7.17* that when the distraction is introduced (when the machine is at point C) a momentary dip in the salience of 'home_present' is caused by the inappropriate selection by the contention scheduler of 'food_present'. However in this scenario, the SAS Modulator is able to immediately overcome the distraction without the assistance of learning, and the machine accomplishes its goal without any deviation. Though this modification appears to correct lesion A, a more potent distraction presented to the machine with the capability of overcoming the SAS Modulation would again 'capture' the robot.

7.4.4. Lesion Study B

Removing the link between the SAS Modulator and the contention scheduler as in *figure 7.13* and *figure 7.14* marked B prevents the SAS Modulator from attenuating inappropriately salient behaviours and results in the path and traces shown in *figure 7.18* and *figure 7.19* respectively.

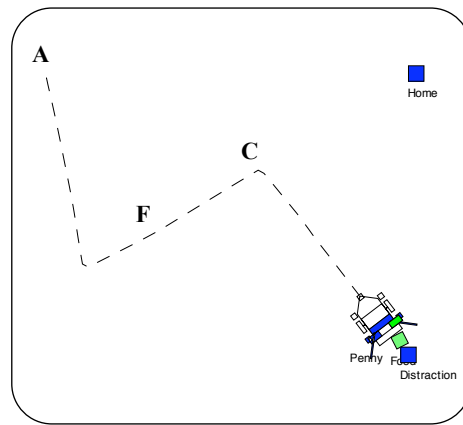


figure 7.18 The path taken by the inexperienced robot Penny during lesion study **B**; the robot starts from point A. A distraction is introduced when the machine is at point C. The neural traces shown in figure 7.19 below, start from F

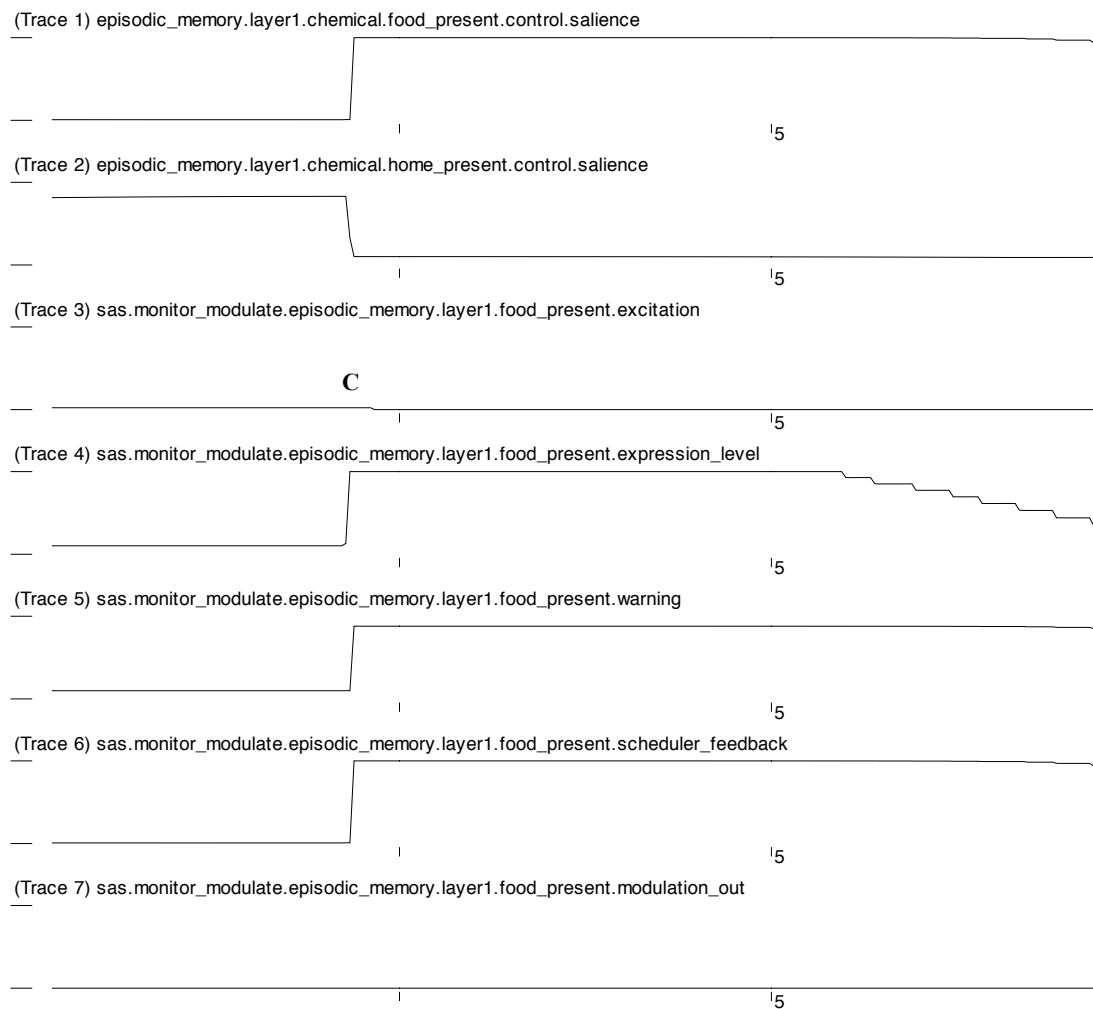


figure 7.19 The associated neural traces of figure 7.18 during lesion study **B**

The robot behaves normally at the beginning of the experiment, however, when the distraction is introduced, although the SAS Modulator generates the modulatory signal in an attempt to suppress ‘food present’ (trace 3), it is not observable at the contention scheduler (trace 7), and so is not able to

attenuate the behaviour's expression. Hence the unintended 'food present' (trace 3) is inappropriately selected by the contention scheduler. Accordingly the machine moves directly towards the source of the distraction as can be seen in *figure 7.19* and traces 1, trace 2 and trace 6.

However as the SAS has generated a modulatory signal, it has also generated an arousal signal to the learning subsystem (hippocampus and limbic system), which as 'touching food' is appropriately expressed is causing 'food present' to be slowly attenuated as can be seen from trace 4.

7.4.5. Lesion Study C

In this third study the SAS is prevented from monitoring the overall salience of a behaviour induced in episodic memory prior to its expression by the contention scheduler (*figure 7.13* and *figure 7.14*, marked C). Hence, the SAS only detects expression of an inappropriate behaviour after it is selected by the contention scheduler for expression. *Figure 7.20* and *figure 7.21* illustrate this failure.

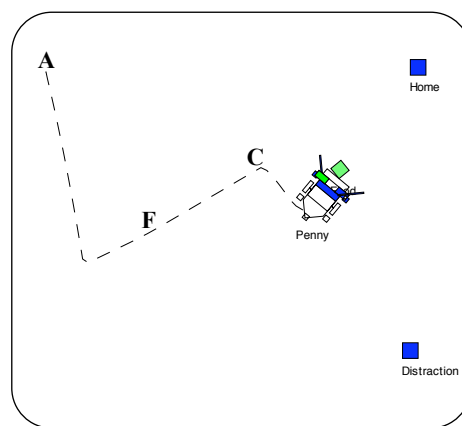


figure 7.20 The path taken by the inexperienced robot Penny during lesion study C; the robot starts from point A. A distraction is introduced when the machine is at point C. The neural traces shown in *figure 7.21* below, start from F

As in the previous experiments, new food is introduced to the inexperienced robot as a distraction whilst the machine is at point C. The distraction results in the salience of 'food present' increasing (the initial rising edge of trace 1) and suppressing the expression of the intended behaviour. As before, this isn't the currently intended sequence in working memory (trace 3) and as in normal operation (*figure 7.7*) the salience of the unfamiliar distraction causes the contention scheduler to express 'food present' (first rising spike of trace 6) which the SAS Monitor detects and generates a warning signal to the SAS Modulator (the first rising edge of trace 5). The SAS Modulator generates a modulatory signal to suppress 'food present' (the rising edge in trace 7) which suppresses the inappropriate behaviour, which is then no longer selected by the contention scheduler (the first falling edge in trace 1) and the intended behaviour is expressed (trace 2).

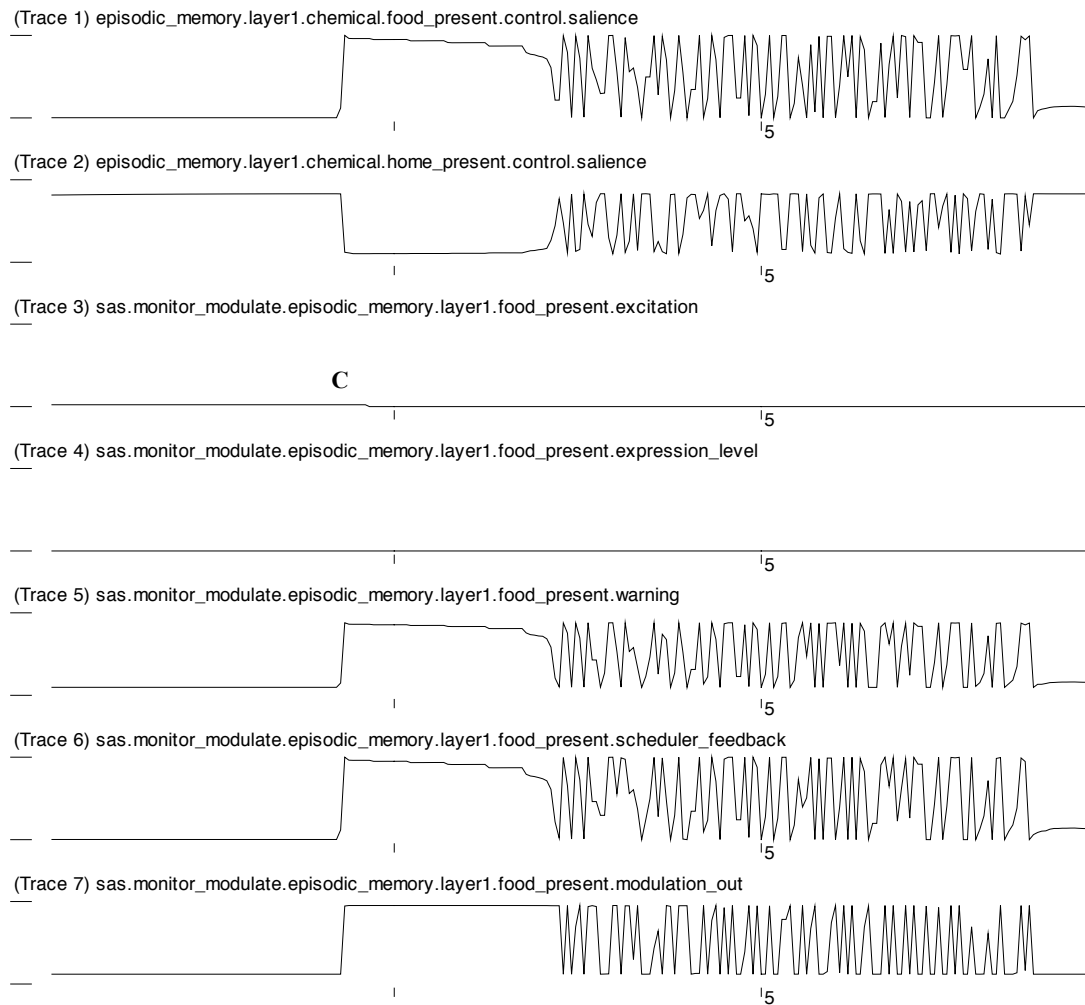


figure 7.21 The associated neural traces of figure 7.20 during lesion study C

However, the lesion introduced to the network prevents the SAS from recognising that the inappropriately expressed behaviour is still strongly salient and is trying to be expressed; accordingly, when ‘food present’ is no longer enabled by the contention scheduler the SAS ceases its modulation of the signal. This results in ‘food present’ again achieving expression via the contention scheduler and the oscillatory cycle of expression and modulation is established (traces 1, 2, 5, 6 and 7). The resulting observed behaviour of the robot (which can be seen by the deviation from a straight line of the machines path in figure 7.20) is that it repeatedly switches between moving towards home and moving towards the distraction (‘food present’ and ‘home present’) moving slowly (a normal machine gets home in just over 5 seconds) in fits and starts in what could be characterised as dithering.

Whilst the machine oscillates between Orient to Food and Orient to Home, the SAS is generating an oscillating arousal signal to the learning subsystem. This eventually manages to increase the inhibitory connection between ‘touching food’ and ‘food present’ so that eventually ‘food present’ is suppressed (the falling edge of traces 1 and 6) and the intended behaviour is re-established.

7.4.6. Lesion Study D

Lesion D (*figure 7.13* and *figure 7.14*, marked **D**) prevents the SAS Monitor being aware of the level of excitation reactively induced in a behaviour as a result of the current pattern of activity in the rest of the system (i.e. combined excitation from working memory, from episodic memory or priming from procedural memory). The effect is that the expressed behaviour monitored by the SAS is considered inappropriate and modulation of that behaviour is induced.

In the first illustration (*figure 7.22*), the lesion is introduced at location **F** somewhere into the experiment. As a reactive response to ‘food present’ is not intended (by working memory) at this point in the experiment, it is suppressed; the traces are exactly the same as in *figure 7.7*. The traces show that the introduction of the distraction induces the inappropriate response to ‘food present’ (trace 1) and that a difference between the intended (trace 3) and expressed (trace 6) responses for ‘food present’ generates a warning signal. The SAS Modulator correctly suppresses its expression (trace 7 of *figure 7.22*). This also generates an arousal signal to the learning subsystem so that over time ‘food present’ is automatically attenuated (trace 4).

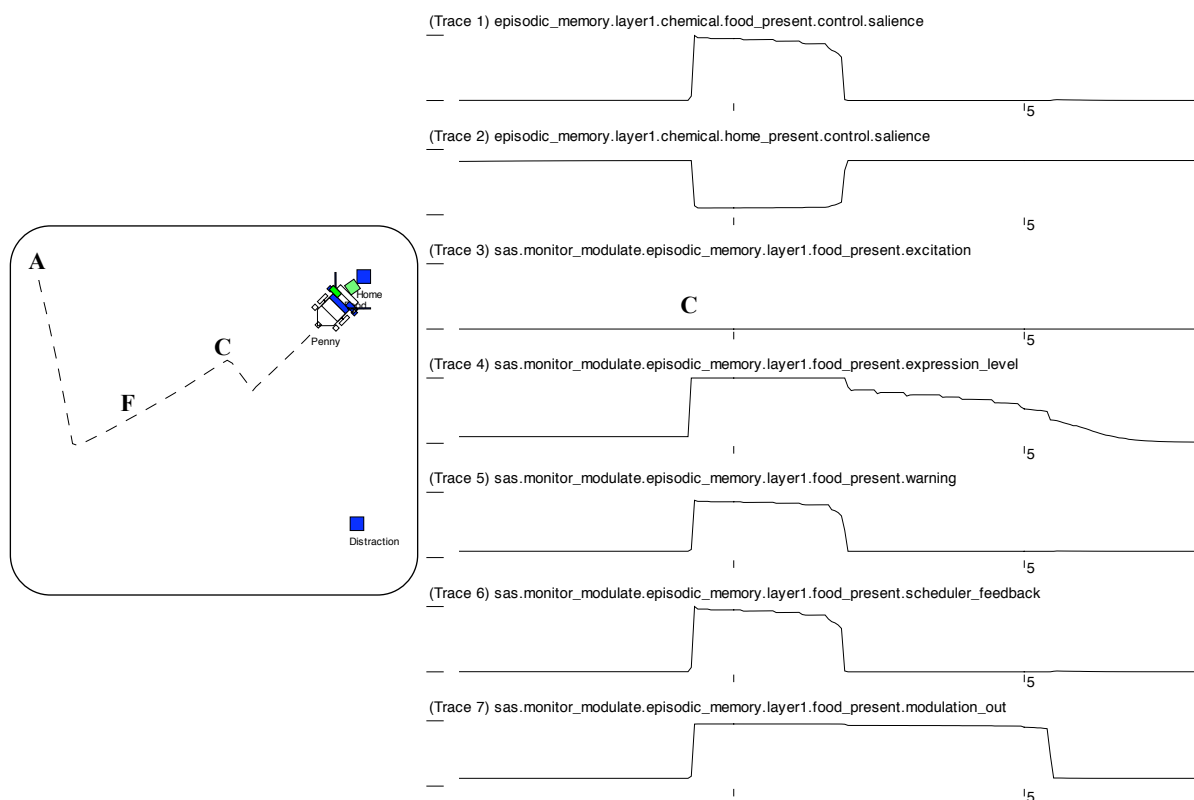


figure 7.22 The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study **D**; the robot starts from point A, the traces start from point F. A distraction is introduced when the machine is at point C (and shown on trace 3)

In a second illustration (traces in *figure 7.23*) the lesion is introduced at the beginning of the experiment (at point A). When the robot exhibits the correct initial response by reacting to ‘food

present' (trace 1) the SAS Monitor (trace 6) induces SAS Modulation to suppress this activity from the onset (trace 7). The effect of the lesion is to induce suppression of an otherwise appropriate behaviour.

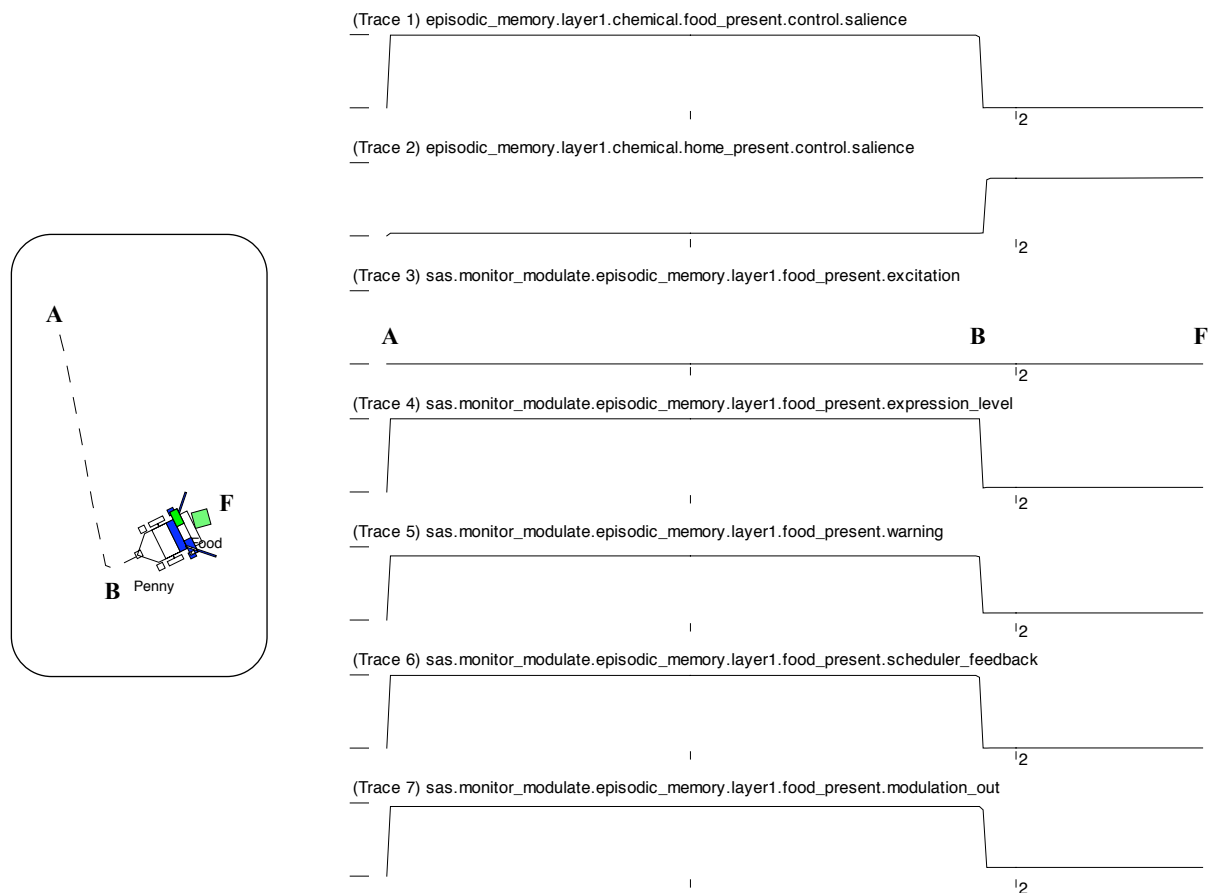


figure 7.23 The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study **D**. The neural traces shown above, start from **A** and end at point **F**.

7.4.7. Lesion Study E

In the final lesion study the SAS is lesioned so that it does not see which behaviours have been selected for expression by the contention scheduler (figure 7.13 and figure 7.14, marked **E**). The traces from this experiment can be seen in figure 7.24.

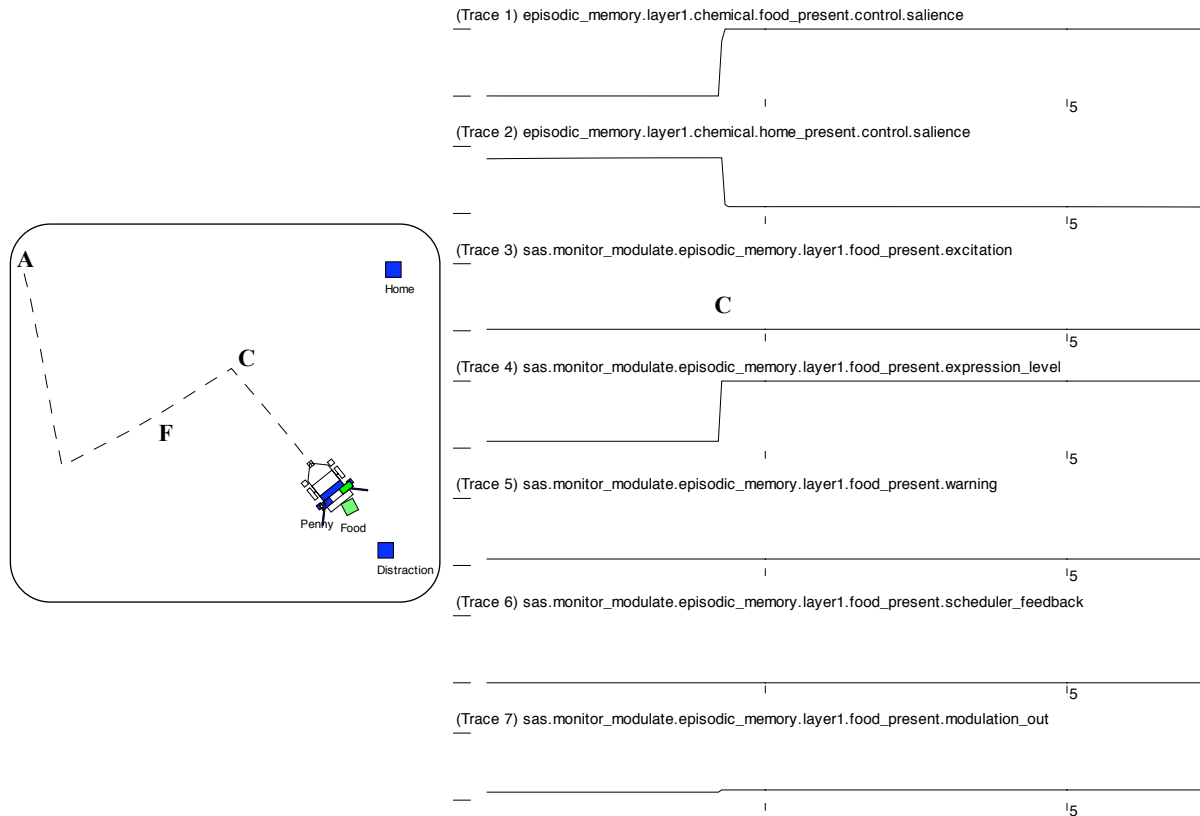


figure 7.24 The path taken by the inexperienced robot Penny and its associated neural traces (on the right) during lesion study E; the robot starts from point A, the traces start from point F. A distraction is introduced when the machine is at point C (also shown on trace 3)

The rising edges of traces 1 to 4 accord with the normal case in which the distraction is introduced. However, on this occasion, no rising edge is seen on trace 6 (as the contention scheduler output to the SAS is lesioned). Accordingly, the SAS view of the currently intended action (trace 3) and the (lesioned) output of the contention scheduler input (trace 6) are the same. Accordingly, no warning signal is generated (trace 5) and hence no suppression of the inappropriate ‘food present’ is generated (trace 7). The small spike in the suppression output seen a second into the trace is from the priming induced by the nodes attempt to secure expression (trace 4), which causes SAS Modulator suppression to persist for recogniser nodes that seek to remain active even when they are suppressed. The resulting behaviour of the machine is simply to move towards the distraction, though unlike lesions B to D, since the SAS Monitor does not detect anything is wrong, the arousal signal for the learning subsystem is not invoked.

7.5. Summary

This chapter has presented the results from simulations designed to illustrate both the normal and lesioned behaviour of the system. A tactical test of attention, based upon distraction, was used to probe the systems responses.

Chapter 8

8. Discussion

Following on from the behavioural studies of an instance of the Attentional Architecture presented in *Chapter 7*, the first part of this chapter seeks to relate the normal behaviour of the machine to the features sought by LaBerge [1999]. Subsequently each lesion is examined and the machines behaviour is compared to a number of known behavioural pathologies that have been recognised as behaviour in human patients suffering lesions to the pre-frontal cortex, an area associated with executive function and attention.

8.1. Normal Behaviour

Before discussing the studies in any detail, an important observation is the evident limitations of external observation of behaviour (the ethological approach). This is highlighted by the fact that different lesions induce similar observed behaviour.

In normal operation, the attentional architecture in its current implementation has been seen to enable a machine to apply attentional effort to deal with unintended reactive responses arising from task-related distraction. Once the machine had learnt to ignore a given level of distraction, attention was only deployed to deal with a new and higher level of distraction. As experience of a range of levels of distraction unfolded, the learning progressed until the expression of unwanted motor action became imperceptible to the observer of the machine itself without the use of the network activity monitor.

This performance certainly complies with the features sought by LaBerge in respect of models of attentional systems [LaBerge, 1999; (see *Chapter 3*)]:

- There is an attentionally driven selection of a less salient but intended task over an alternative response associated with a highly salient, pre-potent, source of distraction. The intention is not derived, either directly or indirectly from the environment, rather it is derived from an intended plan held in working memory. Further, the plan is a set of tasks and exactly which task is to be performed at any given time is determined autonomously as opposed to be specified or explicitly cued in the environment
- The model provides priming of future actions; in episodic memory, nodes (encoding remembered episodes) are primed by links from lower level episodic nodes, nodes in procedural memory and from working memory (indirectly via procedural memory). These nodes are more likely to be activated when the anticipated environmental stimuli relevant to a new task in the intended plan become evident. At the same time, in procedural memory, high-level nodes encoding behaviours prime the associated lower-level behaviours so that they are more likely to be expressed as the associated EM nodes become activated
- Memory is used to maintain task focus, but, importantly, this does not involve persistent attentional effort; the goal is remembered without awareness, serving to provoke an attentional arousal signal when expressed action departs from intended action
- Attentional effort in the normal machine is intermittent. Activity of the SAS as it modulates memory through the contention scheduler is momentary unless the situation requires continued attention

Performance extends to the requirement for autonomous learning based upon an intrinsic signal [Maes, 1994; (see also *Chapter 3*)]:

- Attentional activity (SAS Modulation) is also used to induce affectively mediated learning of the intended goal-related responses so that performance of intended tasks becomes increasingly automatic, freeing attentional resources to deal with new levels or types of distraction

There are two aspects of the normal function of the controller which, at first consideration, appear to depart from expectation. The first of these is that in the relatively experienced machine, the introduction of the distracting stimulus results in momentary expression of an inappropriate reactive behaviour at the machine's effectors as evidenced by the spike in trace 6 of *figure 7.8* and *figure 7.10* from *Chapter 7*. It is not possible to resolve the question of whether this is an artefact of the system or whether it has a root in human responses. In respect of the latter possibility, a related phenomenon may be the so-called Chevreul Pendulum effect (described in [Baars, 1997]) which is observed when a subject is given two contradictory intentions to sustain. Initially, they are asked to hold still a pendulum suspended from their supported hand. This can be done successfully. When asked to will it to oscillate on one or another specified axis, whilst simultaneously maintaining it in a steady state, the pendulum is (usually) induced to oscillate against the will. The small motor actions evident in trace 6 of *figure 7.10* would be consistent with a comparable phenomenon in the robot as the imperceptible motor actions might, eventually, be expected to drive a pendulum.

The second aspect of normal function deserving comment is the fact that, for a given level of distracting stimulus, there is sudden transition from a state in which it demands attention to a state in which it demands no attention, the transition from novice to expert performance seems sudden (falling edge of trace 7 in *figure 7.7* and *figure 7.8*). The orthodox view in respect of skill learning was that acquisition of automatic response was relatively gradual in all subjects [Palmeri, 1999]. However, recently it has been observed that this may be an artefact arising from the aggregation of individual trials in the presentation of results, and that individuals do indeed acquire automaticity in a qualitative step when learning, a task requiring selective attention to task-relevant information [Haider and Frensch, 2002]. This latter observation is consistent with the behaviour of the controller. *Figure 8.1* compares an individual's performance with aggregated results. (Thanks to Dr. Alison Green of the Psychology in Science Group, Department of Biology, The Open University, for these illustrations.)

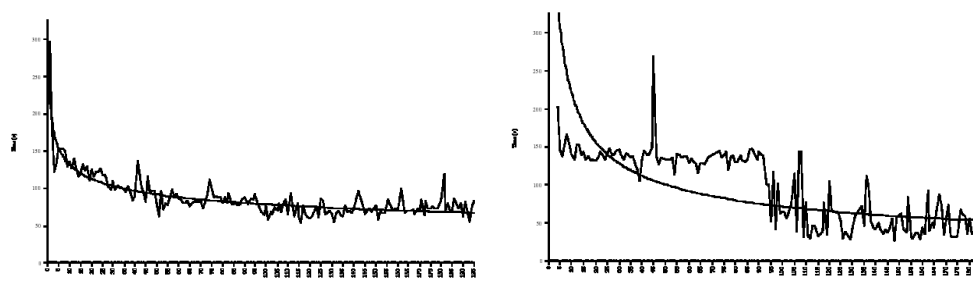


figure 8.1 Improvement in attentional task performance (time taken, *y axis*) with sustained practice (number of trials): aggregated results (left) compared to individual profile (right)

Figure 8.1 raises a question about whether the ‘perfect’ learning exhibited in the simulation (that is, familiarity with all possible levels of a particular distraction) is actually possible in a complex and dynamic environment. In any ‘natural’ context the low probability of encountering all repeatable conditions for task performance in the presence of distraction, even in a laboratory context, is likely to result in some level of departure from the norm in any single trial, as appears evident in the enduring variation in performance seen in *figure 8.1*.

8.2. Lesioned Behaviour

Lesion studies not only reveal the failure modes of the system, they also serve to offer different degrees of support to assumptions made during the detailed design of the architecture and its implementation, that is design decisions taken where there is limited evidence from existing knowledge and research to draw upon. At the level of observable behaviour, there are a number of behavioural pathologies that have been recognised as behaviour in human patients suffering lesions to the pre-frontal cortex, which is associated with executive function and attention. These pathologies of behaviour are rarely clear cut (lesions are seldom highly localised) and often have multiple forms, especially in respect of the self-reported experience of the patient. Examples of such pathologies include the following [Shallice, 1988; Parkin, 1996]:

- Utilisation behaviour is an inability to suppress a (strongly) triggered, but inappropriate behaviour. A typical account is of a patient who reaches for a cup when presented with it even though it has been explicitly agreed by the patient that they will not do so [Shallice, 1988]
- Distractibility (or the making of capture errors) in which the behaviour drifts without notice from one task to another. The classic example is that of the psychologist William James realising he was dressing for bed instead of dressing for dinner, but this form of distraction is widely experienced in different degrees, including in the absence of lesion
- Akinesia is the inability to act, attributable to impairment of the ability to resolve selection between competing behaviours. Some patients report that they are actively willing action but not succeeding
- Stereotypy or perseveration is inappropriate persistence of a behaviour in the face of changed context. It appears to be a failure to notice (bring to awareness) significant cues associated with successful completion of a task or subtask that should result in the expression of a different behaviour. Again, there is a spectrum of patient experience and it has been suggested that everyone experiences some degree of latency between onset of unconscious (somatic) awareness and conscious awareness [Damasio, 1998]

The first two are sometimes considered as different forms or degrees of distractedness; in the first case, the subject initially wills the avoidance of the response, but the stimulus is so strong it is not suppressed, however the person is usually aware of the inappropriate action [Parkin, 1996, p. 226]. In the case of capture errors, the patient is initially unaware of the lapse. All the behaviours illustrated

above can be explained as a failure in error correction (inhibiting a incorrect or inappropriate response) with akinesia and stereotypy also involving a failure to initiate another action.

8.2.1. Lesion A

Lesion **A** removes the stimulus path between the SAS Modulator and the limbic system (hippocampus), effectively preventing attention-based learning. The study revealed that the observed behaviour depended upon the level of the modulatory signal from the SAS Modulator to the contention scheduler. In the first trial this signal was normally low and accordingly failed to suppress the reactively induced response to the distraction. In the second trail the modulatory signal was high and hence the externally observed behaviour of the robot was normal.

It is tempting to consider the latter as the preferred response and to imagine a mechanism that increased the level of the modulatory signal until the intended behaviour was re-expressed. (For example, the level of the signal could be proportional to the attentional effort.) However, a more plausible mechanism for dealing with enduring distraction of which the subject is aware (as is the case in these trials) is that of the SAS Generator. This is the hypothesised mechanism for revising current goals and plans held in working memory.

In both cases the lesioned machines exhibit enduring attentional activity (trace 7 of *figure 7.16* and *figure 7.17*). The ‘experience’ of the second machine would be of constantly resisting the distraction. The first machine would experience an inability not to react to the distraction and would be aware of this, an experience corresponding closely to utilisation behaviour.

(The implications of neither machine being able to learn are considered in *Chapter 9* in a discussion on the possible origins of executive attention.)

8.2.2. Lesion B

Lesion **B** removes the modulatory path from the SAS Modulator to the contention scheduler, preventing the SAS from attempting to directly attenuate an inappropriately expressed behaviour. Accordingly, it is unable to suppress the reactively induced response to the distraction and the robot approaches the second food source. However, the SAS is still able to generate an arousal signal to the learning subsystem. The robot has the original food sources and so the associative response ‘touching food’ is expressed appropriately and so the neuromodulatory learning is causing ‘food present’ to be slowly attenuated, as can be seen from trace 4. Eventually the machine may learn to escape from its distraction, but the inhibitory link between ‘touching food’ and ‘food present’ will be so strong that even mild priming (e.g. anticipation) of ‘touching food’ may inappropriately suppress ‘food present’ resulting in inappropriate behaviour.

It is possible that in an expanded model other active clusters with inhibitory links to ‘food present’ would eventually suppress it and limit, the *inappropriate learning* (damage to the memory systems

caused by the incorrect release of the diffuse neuromodulator) created by this lesion. This behaviour provides a second example of utilisation behaviour.

An extension of this study in which high levels of salience was induced in additional episodic nodes adjacent to (that is with adaptive connections subject to neuromodulatory-based learning, as suggested above for the expanded controller) might offer a means of correcting, or at least attenuating, the effects of the inappropriate learning. Such a strategy might be compared to cognitive behavioural therapy (CBT).

8.2.3. Lesion C

Lesion **C** prevents the SAS Monitor being aware of the salience of recogniser nodes in episodic memory seeking expression via the contention scheduler. Consequently, unintended behaviour is only observed and corrected once it is expressed. The result is the faltering pattern of behaviour as the expressed behaviour switches between moving towards home and moving towards the distraction.

This pattern of behaviour is characteristic of other attentional abnormalities in humans. The first is that of distractedness due to capture. In this simulation the machine responded to a changed environment in a well-learned fashion, turning towards the additional food. In the absence of an intention to perform this behaviour (trace 4 of *figure 7.21*) and the lesioned pathway, the machine did not immediately recognise the changed behaviour. When it did, it dithered between contending behaviours; the nature of the lesion contributing directly to an inability to anticipate its immediate actions.

The second related pathology is akinesia, attributed to impairment of the ability to resolve selection between competing behaviours behavioural [Shallice, 1988]. In the observed machine, the behaviour switching is rapid enough for it to appear to be doing very little; any eventual progress is very slow compared to normal behaviour.

This lesion study provides an opportunity to comment on the time taken to switch tasks, the Stroop effect (see *Chapter 3*). The ‘width’ of the spikes in the traces for this study represent the time taken for the controller to switch focus between two tasks when the stimulus for the switch comes from observation of the environment.

8.2.4. Lesion D

Lesion **D** prevents the SAS Monitor being aware of intended behaviour (in working memory) or ‘excited’ behaviours (from overall activity in the rest of episodic or procedural memory). For such a machine, all expressed behaviour appears unintended. As can be seen from the trace 7, in *figure 7.23*, attentional resources are being used and so there is an awareness of the action. The attentional activity reinforces learning of the suppressed level of behaviour. This corresponds to a known and extreme form of akinesia in which there is a pronounced inability to act, regardless of stimulus.

8.2.5. *Lesion E*

Lesion E prevents the SAS Monitor observing behaviour selected for expression by contention scheduling. Accordingly, it has no immediate awareness of its own expressed actions and SAS intervention is not invoked. Presented with a familiar distraction with a well-learned response, the machine exhibits a classic example of a capture error.

8.3. *Summary*

This account of the performance of the simulated machine has demonstrated good correspondence with the goal of this work: to develop an integrated neural architecture modelled on human executive attention which was capable of supporting autonomous control and intrinsic attention-based learning in a simulated robot.

It has been seen to display features of executive attention as set out at the beginning of the thesis [LaBerge, 1999]:

- The selection of a less salient intended task over a more salient task that has been automatically selected
- Anticipation and priming of tasks by memory subsystems
- The use of memory to maintain task focus and allow the persistence of willed action over automatic behaviour
- The use of attentional effort only when needed

And extended to include:

- Learning appropriate willed actions such that over time their expression becomes increasingly automatic, reducing the need for attentional effort

The task used for the simulated system's evaluation (see *Chapter 7*) conformed to a common 'tactical test' used to clinically assess the monitoring and modulation aspects of executive attentional capabilities in patients [van Zomeren and Spikman, 2003, p. 76]. The reproduction by the lesioned simulation of behavioural pathologies seemingly analogous to those associated with attentional dysfunction in humans lends support to the validity of the architecture and to the elaboration's introduced (often inferred) into the basic functional model as it has developed towards its current implementation.

Chapter 9

9. Concluding Discussion and Future Work

In keeping with the account of ‘structured-case’ given in *Chapter 2*, the Attentional Architecture and its implementation described and examined in the previous chapters constitutes the current conceptual framework for understanding executive attention. This chapter begins by relating the open questions in the field identified and described in *Chapter 1* to the research presented in this thesis. It does so by summarising the nature of the contribution made to each open question together with related future work. The discussion of further work continues by addressing a number of specific issues that arise from the work presented here. Consideration of each issue is initially rooted in a currently perceived problem or limitation of the model but then ventures into less certain or speculative territory in considering ways to take this project forward in a significant direction. The first issue concerns the implications of the model having implemented only the first two sub-functions of the SAS (Monitor and Modulate). The second issue examines the approach taken to induce learning in episodic learning using SAS activity and addresses a possible explanation of how such a mechanism might have developed in humans and animals. The third issue explores issues relating to the scalability of the architecture, which leads to consideration of the anatomical architecture of the SAS. Finally the chapter looks forward to the work which will examine the robustness of this approach to action selection in an embodied machine.

9.1. *Relation of this Research to Open Questions in Cognitive Systems*

Chapter 1 identified a number of open questions relating to behaviour and action selection in biologically based systems which had been posed recently by the UK Government's Foresight Project entitled 'Cognitive Systems' [Morris, Tarassenko and Kenward, 2006, p. 46]. Two related open problems were initially identified [Morris, Tarassenko and Kenward, 2006, p. 46] and, in considering how these problems might be addressed within cognitive and computational neuroscience, two more specific open questions were posed [Barnard and Redgrave, 2006, p. 129]. The (partial) contribution to these questions made within this thesis are summarised in *table 9.1*.

<i>Open Question</i>	<i>Contribution, Additional Questions and Future Work</i>
Developing architectures that allow modules (actions or behaviours) to dynamically and automatically reconfigure themselves [Morris et al.]	<p>The simulation of the Attentional Architecture incorporating the attentionally-derived and neuromodulator-mediated learning signal in episodic memory has demonstrated a mechanism that dynamically reconfigures episodic memory. The effect of this is to increase the capacity of the system to deal automatically with dynamic sources of pre-potent task related distraction. Thus, the impact of common and powerful sources of distraction on performance are diminished</p> <p>The work presented here has pointed to, but not yet implemented, a learning mechanism capable of reconfiguring procedural memory. Currently, behavioural change is constrained to re-sequencing a number of pre-determined motor actions as opposed to developing wholly new motor actions</p>
Enabling agents to generate novel behaviours and, through learning, integrate these new behaviours with its functioning set of behaviours and its strategic objectives [Morris et al.]	<p>At the present stage of the development, the architecture does not support the autonomous generation of novel behaviours. However, it does provide a partial solution to this problem in that it provides a means for representing novel behaviours in working memory and demonstrates the capacity of the reactive system to progressively learn such a behaviour once formulated and held in working memory for an extended period of time. This partial solution provides a platform that allows the generation and integration of novel behaviour to be investigated</p>

In the context of Norman and Shallice's model of executive attention, a mechanism for implementing the SAS generator is required. This is likely to involve a number of distinct mechanisms rather than a single, high-level planning system. For example, simple behavioural strategies for dealing with problems might include reverting to the previous behaviour or initiating the next behaviour in a sequence. Such strategies require only modest alterations to the information processing in the existing architecture. At the next level of sophistication a backward chaining mechanism may be used, such as that described by Garagnani et al. [2002] where episodic memory is searched for contextually relevant behaviours

How is action selection influenced or determined by attentional mechanisms working upon information about current external states of the environment and internal states of the agent?
[Barnard and Redgrave]

The properties of the Supervisory Attention System, as designed and simulated, provide a comprehensive, well developed, and qualitatively evaluated working hypothesis to this question. Action specification and expression is seen to be modulated by an attentional system using episodic memory to represent external states and working memory and procedural memory to represent the reactive and deliberate intentions of the system, respectively

There are a number of questions posed by the work of this thesis in relation to the mechanism for attentional modulation of behaviour. The most immediate concerns the means by which the appropriate level of modulation is determined (see *Chapter 7, section 7.4.3* for the relevance of this). Currently, within the Attentional Architecture it is proportional to the salience of the inappropriately expressed behaviour, but the question arises as to how the level (rate) of proportionality is determined autonomously, e.g., through experience

Beyond the question above, there are other possible architectural configurations to be considered, e.g., the possibility of modulatory connections directly into procedural memory

How does information in memory
interact with those current states?
[Barnard and Redgrave]

The architecture of the attentional system establishes the information structures and pathways necessary for interaction and the neural simulation provides one expression of the dynamics of the interaction (in this instance, the underlying dynamics are, by design, chaotic). The resulting behaviour emerges as observable and robust patterns of response from a highly complex and integrated neural information processing system

At the architectural level, an interesting question is whether the architecture presented here can be extended in a simple way (i.e. using existing components and mechanisms) to elaborate how working memory could access states such as hunger and thirst, and beyond these, other physiological states associated with stress, anxiety, pain and pleasure. The proposed line of development is to treat the neural structures and mechanisms that sense these states as analogous to episodic memory, forming an episodic memory for affective associations. Such a structure would interface to working memory in exactly the same way as behavioural episodic memory

table 9.1 The (partial) contribution to the open questions posed in *Chapter 1* made within this thesis

9.2. Behavioural Evaluation and the SAS Generator

The qualitative evaluation of behaviour presented in the previous chapter was based upon the current implementation of the architecture. There are two particular challenges to interpreting and evaluating this behaviour; the first at the level of external observation, the second at the level of network activity traces.

At the level of observation, a problem is posed by the absence of one of the three elements of the SAS specified by Norman and Shallice [1986] and elaborated by Burgess and Shallice [1993] and later again by Shallice and Burgess [1998], namely the SAS Generator. The SAS Generator is responsible for formulating new goals and intentions (represented in working memory) when routine modulation does not result in the intended behaviour correction. In considering the observed behaviour of the machine in these studies, it seems evident that such an additional mechanism would ordinarily have been invoked on a number of occasions. Specifically, it would have been invoked on any occasion where the SAS Modulator remained active in the presence of enduring inappropriate behaviour.

In lesion study **A**, two variants of behaviour were seen which were determined by the level of modulation applied by the SAS Modulator. In both cases, the SAS continued to apply modulation. In the discussion of *Chapter 8* the notion of devising a learning signal that assured behavioural modulation was rejected. The reason being that such a mechanism would tend to ensure progressive compliance with any currently intended behaviour. It would not leave room for a mechanism we associate with the SAS Generator that involves consideration of whether the willed behaviour continues to be the optimal behaviour, or whether a different behaviour should be substituted. The presence of such a mechanism is clearly part of ‘normal’ behaviour’ and observation of behaviour where it is precluded is more difficult to analyse. It is as if there is an initial lesion upon which the subsequent lesions are superimposed. Addressing this absence is not a trivial task in a neural system (see *section 9.2.1*, below).

In respect of interpreting the network activity traces, the problem posed is the need to visualise the highly complex patterns of activity arising over time in a densely recurrent network. As an example of the complexity one can consider lesion **D**. In the current architecture, this is presented as preventing the SAS Monitor from observing the degree to which a reactive behaviour is induced by activity across the whole network. However, this is not entirely accurate as the network-wide activity also influences the level of activity of each node in episodic memory, and this is observed by the SAS Monitor in terms of the node’s activity as presented at the contention scheduler. If the behaviour is then expressed by the contention scheduler, the SAS Monitor observes this too. Thus, interpreting the traces requires detailed consideration of the flow of information in the network. Of course, this is a fundamental problem in conventional neural imaging studies. This latter problem presents particular difficulty in developing studies of subtle compound lesions, for example, by combining lesions **A-E**. (Although some combinations are trivial in their effects, such as any that includes **A+B**, which simply eliminated the SAS.)

9.2.1. Further work I

In their elaboration of the SAS Generator Shallice and Burgess [1998] present a ‘box and arrow’ account featuring a large number of sub-processes, itself as rich as the original SAS model (*Chapter 3*). Its role is to generate plans, respond to impasse events and generally organise behaviour in such a manner that goals and intentions are met. Nothing is actually known about any of the processes the SAS Generator is supposed to carry out or how they may be implemented neurally, though at least one model of part of a neural planning system does exist [Garagnani, Shastri and Wendelken, 2002]. There may be scope for extending the current model to include this further subsystem.

A method of evaluating possible courses of action has been proposed by Shanahan [2006]. However, his model features a second order system that largely duplicates the reactive control system in order to explore forward behaviour ‘off-line’. There is no real evidence either to support or refute the existence of such a system, but this level of duplication is not common in neural systems. The model presented in this thesis presents potential for an alternative approach to evaluating plans. It relies upon the

predictive links from procedural to episodic memory as proposed by Frith [1998]. Currently, these links are only used to prime episodic memory for anticipated future events in the world. They create a cyclic processing structure within the three memory subsystems, episodic memory to working memory, working memory to procedural memory and procedural memory back to episodic memory. If the SAS was able to hold lower levels of episodic memory stable whilst simultaneously inhibiting lower levels of procedural memory (ensuring that actions were actually expressed), the SAS Generator could cycle through memory [Marshall, 2007] to evaluate proposed actions.

9.3. Attention, Autonomous Learning and Selective Advantage

The attention-based learning induced by the SAS Modulator is realised through adjustments to the weights on connections between active and suppressed nodes in episodic memory (inhibitory links from active nodes to SAS suppressed nodes only increase in weight). This learning is potentiated by the simulated concentration of a neuromodulator which is proportional to SAS activity. This mechanism provides for increased automaticity of response as initially novel situations become more familiar or sustained attention is applied for a sufficient period. However, the attentional effort and learning that occurs due to a novel or unfamiliar environment can, if that environment continues to be novel over an extended time period, cause over-learning with some undesirable consequences:

- Robust SAS induced learning requires several nodes with inhibitory links to the inappropriately expressed node to be active. If few nodes are active, the subsequent learning may be very specific about the environment or the internal state of the machine and less adaptable given other states or sensory conditions
- A lack of robust learning, coupled with inappropriate or weak expression (as recogniser nodes seek to suppress other nodes) will lead to the contention scheduler inappropriately choosing nodes for expression, requiring increased ‘attention’ from the SAS Modulator and thus more learning

A more subtle learning mechanism is needed. One possible approach is to adopt a refinement to the above learning scheme suggested by sleep studies in humans and animals. Echoing the learning mechanism described above, which is implemented in the current system, Tononi and Cirelli [2003; 2006] propose that learning during an aroused, wakeful state, increases synaptic strengths across the cortex. Controversially they suggest that the role of sleep (and specifically thalamo-cortical oscillations during non-rapid eye movement sleep) is to “downscale synaptic strength to a baseline level”. If included in the current system, the effects of such downscaling would be to lower the weights of the inhibitory links over time in a way that led to ‘forgetting’ adaptations which were made through sustained attentional effort on a single occasion or were very rarely encountered. Although the SAS Monitor and Modulator would have to do additional work to refresh automatic behaviour that had been partially forgotten, potentially more (or different) recogniser nodes would be active when learning to suppress the inappropriate expression on subsequent occasions, creating more robust

automatic behaviours and lessening overall attentional effort in correcting previously over-learned episodes.

Consideration of why such an approach might be appropriate is part of a somewhat larger question posed by LaBerge [1999] among others: why might executive attention, as a phenomenon, have emerged? This question can be considered in two parts: ‘why’ and ‘how’.

A perspective on ‘why’ is provided by Broadbent’s [1958] filter theory of attention which proposed that because cognitive resources are limited then attention is needed in order to manage access to these resources. A related but larger point is made by Aleksander and Dunmall [2003] who suggest that attention is necessary when the building and maintaining of an internal representation of the perceivable world cannot be done in parallel. This perspective reflects the fact that brains are finite and the information available to them is governed by location and orientation, and by time; it provides an answer as to why attention has emerged.

To answer the question about ‘how’ it might have arisen it is appropriate to ask how executive attention might contribute to selective advantage. The current model of learning in the Attentional Architecture provides for attention-based learning in episodic memory (and indirectly, in procedural memory) so that initially novel perceptual patterns become more familiar, and the associated actions become capable of automatic expression without attentional involvement. The learning in the architecture arises from an intrinsic and innate mechanism that assumes a set of initial (basic) capabilities ‘at birth’. In a dynamic and unfamiliar world, the attention-based learning operates so that agent is capable of learning to distinguish between the familiar and the unfamiliar.

- When the unfamiliar is detected, executive attention allows scarce, expensive (in terms of resource) cognitive mechanisms to be brought to bear on the problem, interrupting or abandoning routine behaviour
- Frequent or enduring use of executive attention on the same or closely related problems reinforces automatic response, freeing executive attention to address other issues of novelty

It is plausible to suggest that in a dynamic and stochastic environment there would be natural evolutionary pressure in favour of organisms that could learn to distinguish between the novel and the familiar so that they can focus limited cognitive resources on the novel and treat the familiar in a routine, unattended, fashion. Without such learning, an agent is destined to respond attentionally to all departures from expectation, whether they pose a threat or not. It is not obvious that selective advantage derives from acquiring an automatic response to single or very rare events that require sustained attention and thus induce learning, especially as the capacity of episodic memory appears to be limited. This underpins the value of the extension to the learning mechanism described above,

which biases automatic learning towards more frequently encountered contexts and promotes forgetting of rarely met contexts.

A distinctive element of the Attentional Architecture described in this thesis is the inclusion of the affective (limbic) system in learning. If the system were to be extended to include elements of the endocrine system (which is closely associated with activity of limbic structures) then it may even be possible to establish an addictive underpinning of learning. Such a system would be driven by novelty. Its neophilic behaviour would be characterised by 'adventurousness' and 'curiosity'.

9.3.1. Further work II

In extending the affective learning system, one possible line of development is to incorporate the functionality of another structure within the limbic system, the amygdala [Beeman et al., 1995] for which a computational model has already been designed [Morén and Balkenius, 2000] and elements of the endocrine system [Dorman and Gaudiano, 1995]. The inclusion of these systems should allow the Attentional Architecture to generate and respond to additional neuromodulatory substances, including those that provide a basis for the hypothesised 'addiction to novelty'.

9.4. Scalability and the Architecture of the SAS

9.4.1. Scalability

The current implementation of the architecture has very few sensors, very few actuators and a very limited repertoire of behavioural responses. One obvious question to be addressed is the problem of scaling up such an implementation.

The scalability of the sub-systems that contribute to reactive behaviour can be related directly to the anatomical architecture of the functional subsystems of the human brain. This is particularly true of the organisation of the sub-units which map perception to episodic memory and effectors to procedural memory. Their organisation reflects the somatotrophism of the human sensory and motor cortex. Increasing the number of sensors or effectors can be compared to increasing the number of sensory or effector organs of a human. This would be reflected in changes to the somatotrophic map of the sensory and motor cortex. Thus, the key factor in scaling of the perceptual system (the lowest level of episodic memory), and the systems that map directly onto effectors (the lowest level of procedural memory, the thalamic gates, and the behaviour contention scheduler), is the number of sensors and effectors. For each sensor added to an existing sensory organ there is an additional element of the perception layer and for each effector added to an effector organ there is an additional element of procedural memory, the thalamic gateway and the behavioural contention scheduler. Where new sensor or effector organs are added, a corresponding branch of episodic or procedural memory is added, in line with the structure of those systems as described by Fuster [2003].

Scaling of reactive responses, which requires extension of both episodic and procedural memory is more expensive. Both episodic and procedural memory are layered and the layers increase in breadth as the depth of the hierarchy increases. At each layer, the hierarchies are connected in both directions and the connections from episodic to procedural memory are subject to contention scheduling, so the size of this sub-systems increases accordingly.

Whilst it was possible to relate the scaling of perceptual and effector systems to the macro neuroanatomy of the brain, this is not possible for the SAS as there is little consensus on its organisation at an anatomical level. In the architecture developed here, the Monitor and Modulate sub-functions of the SAS are organised in a way that reflect the structure of the reactive system that is monitored and modulated. This means that these functions are realised through a structure that is very closely related to the contention scheduler and the thalamus. Its size is governed by the size of the representation in working memory of the uppermost layers in episodic and procedural memory.

The above discussion of scalability can be summarised as follows:

- The number and complexity of sensors dictates the number of episodic memory hierarchies
- The number and complexity of actuators dictates the number of procedural memory hierarchies
- The size of episodic, procedural and working memory are interrelated
- The number of shared information paths between episodic, working and procedural memory and the number of actuators defines the size and complexity of the contention schedulers and the thalamus gates
- The size of the memory subsystem together with the size and complexity of the contention schedulers defines the number of SAS Monitor and Modulator elements

Of these, the most significant are the number and complexity of the sensors and actuators, the overall size of episodic memory and the design of the shared information paths. Once these parameters are fixed the number of recogniser nodes, connections between them and SAS Monitor and Modulator units can be estimated.

9.4.2. *Architecture of the SAS*

Although executive attention is traditionally associated with the frontal and pre-frontal cortex, the identification of a number of SAS sub-functions allows for its functionality to be distributed rather than localised. In the detailed development of the current model, it was very straightforward to implement both monitoring and modulation in a distributed fashion, as described above. Thus, the modelling has tended to follow Fuster [2003] who suggests that these executive functions are indeed distributed. Given the weight of evidence supporting the frontal areas as being implicated in planning

and executive control of attention, it thus seems likely that the frontal areas are the locus of the SAS Generator function.

9.4.3. Further Work III

The immediate key to scaling of the architecture is to develop an approach to designing behaviours and the shared information paths that connect episodic, working and procedural memory together. One possible way forward is to use Baerends [1976] diagrams. These ethological tools are useful in modelling how higher level behaviours are composed of lower level behaviours.

9.5. Robustness of Behaviour and Development of an Embodied Machine

The evaluation of the architecture presented in this thesis has used qualitative observations to explore and understand the behavioural properties of a normal and a lesioned simulated machine. The observed behaviours and the associated traces presented in *Chapter 7* were illustrative of qualitative behaviour that consistently emerged from individual runs of the simulation for which the network evaluation or algorithm ensures a chaotic sequence of network states (see *Chapter 6, section 6.3.1* and *section 6.3.2*). A quantitative evaluation of the robustness of architecture's performance requires a statistical investigation of the architecture's performance. As with the qualitative studies, the analysis must operate at two levels: the externally observable behaviour of the simulated robot and the interpretation of the traces which reveal the detailed functioning of the controlling architecture in both its normal and lesioned states.

9.5.1. Statistical Analysis of Observable Behaviour

Robustness concerns the reproducibility of some reference behaviour. In the case of foraging, the first (and trivial) measure of robustness is effectiveness, which is simply concerned with investigating the extent to how successful the foraging behaviour is, independent of the initial locations of food, home or the location and orientation of the machine. The most elementary measure of robustness is the percentage rate of success in locating and subsequently delivering food to home over a statistically meaningful number of trials (at least 30 and preferably circa 100) with each trial involving random placing of the machine, the food, and home (subject to some minimal separating distance so that no two elements are co-located). A slightly more demanding measure of robustness would consider efficiency. In the first instance, efficiency can be considered in terms of the difference between the length of the path actually taken and the length of the optimal path (minimal Euclidean distance) relative to the length of the optimal path. With these measures of robustness, the descriptive statistics of interest are the proportion of trial successes, and the mean and variance of the task efficiency (as defined). Analysis of the data will involve characterisation of the distribution of the efficiency measure, and its mean and variance. Given that the measure of efficiency proposed takes into account the overall path length in any one trial, it would be relevant to explore the extent to which the statistics showed a residual dependence on path length.

The above evaluation needs to be repeated for normal function in the presence of distraction. The salience of a distraction is a function of both its distance from the robot and of the degree to which the robot has learnt to ignore that distraction (by virtue of previous exposure). Both of these can be treated as independent variables against which the three statistics (rate of success, mean efficiency and variance of efficiency) can be evaluated.

Finally, it is appropriate to evaluate efficiency in the light of the individual lesions, **A-E**, as described in *Chapter 7*. For those lesions resulting in behaviour suggestive of akinesia (e.g., through repeated task switching), measurement of path length is not wholly sufficient to capture the impact of the lesion. In this case there needs to be an additional comparison between normal and lesioned controllers which examines the time taken to complete the path. A more controlled comparison between normal and lesioned machines can be made if an initial set of randomly generated trial configurations (location of machine, food, home, distraction, degree of prior learning) is stored and used for the comparison.

9.5.2. Statistical Analysis of Control Traces

When examining individual control traces, the issue is not directly task performance and efficiency but the reproducibility of the patterns of control behaviour and some of the associated trace parameters. Most of the features of interest in the traces involve significant changes in the levels of activity in nodes (graphically, these appear as step changes, though they are, in fact, rapid but continuous changes). The timing of these changes is sensitive to many factors including, the precise sequence of network state evaluation in any one run, the history of the machine, the exact locations of all elements in any trial. In *Chapter 8, section 8.1* it was shown that the simple averaging of such traces could reinforce misleading interpretations. Such problems occur with a large range of physiological/neurological signals. One common approach is to contrive an externally triggered event common to all traces which is used as a reference point in time. Introduction of the source of distraction is the most obvious event in this context and the timing of this event can be recorded. With such a reference event, there are a number of ways to progress. Graphically, it is possible to superimpose a number of traces and to make judgements as to gross features such as ‘bunching’ of events and their relative endurance. This is common practice when seeking to demonstrate reproducibility. Statistical analysis tends to focus on the latency of onset and duration of events relative to the reference signal. In this context, this allows analysis of the perhaps the most important property of the control architecture: attentional effort when dealing with distraction as a function of prior experience, and the impact of the different lesions upon this.

9.5.3. Developing an Embodied Machine

For developers of autonomous neurally controlled robots, it has long been recognised that control systems developed in a simulated environment perform poorly in embodied machines, a major contributory factor being the inherent noisiness of signals in the real world (Miglino et al., 1995). In

anticipation of deploying the Attentional Architecture in an embodied machine it is appropriate to examine its performance when signal noise is simulated. The properties of noise in the real world are too complex to model (for sonar signals it would involve taking into account all of the reflective surfaces, both static and dynamic; for chemical gradients the effects air flow and consequent non-uniform diffusion). However, a common component of real world noise is a Gaussian signal error with mean zero and non-zero variance and so the addition of such an error signal to sensors of the simulated robot provides a useful first approximation to real world noise. The first independent variable of interest in respect of the effect of noise on performance is the variance of the Gaussian noise. Initially, this can be assumed to be independent of the distance between the sensor and the source of the signal; alternative assumptions to be investigated subsequently are that the variance may be proportional to either or both of signal magnitude and distance from source.

For the Attentional Architecture, the introduction of such signal noise raises very specific issues. At a sufficiently high level, the noise will induce either a false negative and false positive episodic association with any or all of the food, the home location or the distraction. For food and home both of these errors will have consequences for the efficiency of the machine (as measured above). Repetition of the analysis discussed in *section 9.5.1* and *section 9.5.2* in the presence of noise will establish more precisely how noise impacts effectiveness and efficiency.

The issues relating to distraction are less straightforward. A false negative fails to distract and, as the distraction in this context is not threatening or problematic in any way and it can be discounted. A false positive in respect of a distraction may be expected to present more interesting issues of behavioural evaluation, especially if it is sufficiently salient to induce a switch in automatic behaviour. Even if momentary, such a switch in behaviour should lead to attentional modulation of the erroneous behaviour and induce an associated learning effect that can be expected to diminish the susceptibility of the machine to further false positives with similar salience or less. Thus, the machine may demonstrate improvement in its efficiency over time given sufficient levels of signal noise for a signal of given salience. The statistic of interest here is the frequency of change of behaviour and its relationship to both the magnitude of the noise and the duration of the exposure.

9.6. Summary

This chapter has summarised the relationship of the work presented in the thesis to some recently highlighted open questions in this field. It has discussed some of the immediate issues relating to the current model and its implementation. In doing so, it has indicated some the main lines of exploration that lie ahead and to some of the adjacent research which might be built upon. Finally, it has pointed to ways in which a statistical assessment of the systems robustness can be examined, including the introduction of simulated signal noise.

References

- [Aggleton and Pearce, 2001] Aggleton, J. P. and Pearce, J. M. (2001). Neural systems underlying episodic memory: insights from animal research. In Baddeley, A., Conway, M., and Aggleton, J., editors, *Episodic Memory*, chapter 12, pages 204–231. Oxford University Press.
- [Aleksander and Dunmall., 2003] Aleksander, I. and Dunmall., B. (2003). Axioms and tests for the presence of minimal consciousness in agents. *Journal of Consciousness Studies*, 10(4-5):7–18.
- [Alexander, 1995] Alexander, G. E. (1995). Basal ganglia. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, chapter Part III: Articles, pages 139–144. MIT press.
- [Allport, 1985] Allport, D. A. (1985). Distributed memory, modular systems and dysphasia. In Newman, S. K. and Epstein, R., editors, *Current Perspectives in Dysphasia*. Churchill Livingstone, Edinburgh.
- [Allport et al., 1972] Allport, D. A., Antonis, B., and Reynolds, P. (1972). On the division of attention: A disproof of the single channel hypothesis. *Quarterly Journal of Experimental Psychology*, (24):225–235.
- [Almásson and Sporns, 2001] Almásson, N. and Sporns, O. (2001). Perceptual invariance and categorisation in an embodied model of the visual system. In Webb, B. and Consi, T. R., editors, *Biorobotics: Methods & Applications*, chapter 7, pages 123–143. MIT Press.
- [Baars, 1993] Baars, B. J. (1993). How does a serial, integrated and very limited stream of consciousness emerge from a nervous system that is mostly unconscious, distributed, and of enormous capacity? In Bock, G. and Marsh, J., editors, *CIBA Symposium on Experimental and Theoretical Studies on Consciousness*, pages 282–290. John Willey and Sons, London.
- [Baars, 1997] Baars, B. J. (1997). *In the Theatre of Consciousness*. Oxford University Press.
- [Baars, 2003] Baars, B. J. (2003). The global brainweb: An update on global workspace theory. guest editorial. *Science and Consciousness Review*.
- [Baars and Franklin, 2003] Baars, B. J. and Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Science*, 7(4):166–172.
- [Baddeley, 1987] Baddeley, A. (1987). *Working Memory*. New York, Oxford University Press.

- [Baddeley, 1993] Baddeley, A. (1993). Working memory or working attention? In Baddeley, A. and Weiskrantz, L., editors, *Attention: Selection, Awareness, & Control*, chapter 8, pages 152–170. Oxford University Press.
- [Baddeley, 2002] Baddeley, A. (2002). The concept of episodic memory. In Baddeley, A., Conway, M., and Aggleton, J., editors, *Episodic Memory*, chapter 1, pages 1–10. Oxford University Press.
- [Baddeley and Hitch, 1974] Baddeley, A. D. and Hitch, G. J. (1974). Working memory. *Recent Advances in Learning and Motivation*, 8:47–89.
- [Baddeley and Logie, 1999] Baddeley, A. D. and Logie, R. H. (1999). Working memory: The multiple-component model. In Miyake, A. and Shah, P., editors, *Models of Working Memory*, chapter 1, pages 1–27. Cambridge University Press.
- [Baerends, 1976] Baerends, G. P. (1976). The functional organization of behaviour. *Animal Behaviour*, 24(4):726–738.
- [Barnard and Redgrave, 2006] Barnard, P. and Redgrave, P. (2006). Action. In Morris, R., Tarassenko, L., and Kenward, M., editors, *Cognitive Systems: Information Processing Meets Brain Science*, pages 121–137. Elsevier Academic Press, London.
- [Beeman et al., 1995] Beeman, M., Ortony, A., and Monti, L. A. (1995). Emotion-cognition interactions. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, chapter Part III: Articles, pages 981–984. MIT press.
- [Beer et al., 1990] Beer, R. D., Chiel, H. J., and Sterling, L. S. (1990). A biological perspective on autonomous agent design. *Robotics and Autonomous Systems*, 6:169–186.
- [Boehm, 1988] Boehm, B. W. (1988). The spiral model of software development and enhancement. *Computer*, 21(5):61–72.
- [Braitenberg, 1984] Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. MIT Press.
- [Broadbent, 1958] Broadbent, D. E. (1958). *Perception and communication*. Pergamon Press.
- [Brooks, 1986] Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal Of Robotics And Automation*, RA-2:14–23.

- [Brooks, 1990a] Brooks, R. A. (1990a). The behavior language; user's guide. AI Memo 1227 April, M. I. T. Artificial Intelligence Laboratory,.
- [Brooks, 1990b] Brooks, R. A. (1990b). Challenges for complete creature architectures. In Meyer, J.-A. and Wilson, S. W., editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90)*. MIT Press.
- [Brooks, 2003] Brooks, R. A. (2003). *Robot: The Future of Flesh and Machines*. Allen Lane, The Penguin Press.
- [Brooks et al., 1988] Brooks, R. A., Connell, J. H., and Ning, P. (1988). Herbert: A second generation mobile robot. AI Memo 1016 January, M. I. T. Artificial Intelligence Laboratory.
- [Brown and Chattarji, 1995] Brown, T. H. and Chattarji, S. (1995). Hebbian synaptic plasticity. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, chapter Part III: Articles, pages 454–459. MIT press.
- [Carroll and Swatman, 2000] Carroll, J. M. and Swatman, P. A. (2000). Structured-case: a methodological framework for building theory in information systems research. *European Journal of Information Systems*, 9:235–242.
- [Cliff and Ghanea-Hercock, 2006] Cliff, D. and Ghanea-Hercock, R. (2006). Autonomous agents. In Morris, R., Tarassenko, L., and Kenward, M., editors, *Cognitive Systems: Information Processing Meets Brain Science*, pages 48–58. Elsevier Academic Press, London.
- [Cliff et al., 1993] Cliff, D., Harvey, I., and Husbands, P. (1993). Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):73–110.
- [Cohen et al., 1990] Cohen, J., Dunbar, K., and McClelland, J. (1990). On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychological Review*, 97(3):332–361.
- [Cooper, 2003] Cooper, R. (2003). Organisation and disorganisation of complex sequential activities. In Schaub, H., Detje, F., and Dörner, D., editors, *Proceedings of the Fifth International Conference on Cognitive Modelling. Universitäts-Verlag, Bamberg, Germany*, pages 51–56.

- [Cooper and Shallice, 1997] Cooper, R. and Shallice, T. (1997). Modelling the selection of routine action: Exploring the criticality of parameter values. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 131–136. Cognitive Science Society, Lawrence Erlbaum Associates.
- [Cooper and Shallice, 2000] Cooper, R. and Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, (17):297–338.
- [Cowan, 1999] Cowan, N. (1999). An embedded-process model of working memory. In Miyake, A. and Shah, P., editors, *Models of Working Memory*, chapter 3, pages 62–101. Cambridge University Press.
- [Damasio, 1998] Damasio, A. R. (1998). The somatic marker hypothesis and the possible functions of the prefrontal cortex. In Roberts, A. C., Robbins, T. W., and Weiskrantz, L., editors, *The Prefrontal Cortex: Executive and Cognitive Functions*, pages 36–50. Oxford University Press, Oxford.
- [Dennett, 1998] Dennett, D. C. (1998). *Brainchildren*. Penguin Books.
- [Dorman and Gaudiano, 1995] Dorman, C. and Gaudiano, P. (1995). Motivation. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, chapter Part III: Articles, pages 981–984. MIT press.
- [Dudek and Jenkin, 2000] Dudek, G. and Jenkin, M. (2000). *Computational principles of mobile robotics*. Cambridge University Press, New York, NY. (in press).
- [Duncan, 1993] Duncan, J. (1993). Selection of input and goal in the control of behaviour. In Baddeley, A. and Weiskrantz, L., editors, *Attention: Selection, Awareness, & Control*, chapter 3, pages 53–71. Oxford University Press.
- [Edelman, 2006] Edelman, G. M. (2006). From Brain Dynamics to Consciousness: A Prelude to the Future of Brain-Based Devices. Keynote Presentation, IBM Almaden Institute 2006, <http://www.almaden.ibm.com/institute/resources/2006/Almaden> Institute Gerald Edelman.ppt.
- [Engle et al., 1995] Engle, R. W., Conway, A. R. A., Tuholski, S. W., and Shisler, R. J. (1995). A resource account of inhibition. *Psychological Science*, (6):122–125.
- [Fain, 1999] Fain, G. L. (1999). *Molecular and Cellular Physiology of Neurons*. Harvard University Press.

- [Ferrell, 1993] Ferrell, C. (1993). Robust agent control of an autonomous robot with many sensors and actuators. AI Technical Report 1443 May, M. I. T. Artificial Intelligence Laboratory.
- [Fragopanagos and Taylor, 2004] Fragopanagos, N. and Taylor, J. (2004). The effect of emotional bias in attentional processes. In *International Joint Conference on Neural Networks*, volume IJCNN04, pages 507–512. IEEE.
- [Franklin et al., 2005] Franklin, S., Baars, B. J., Ramamuthy, U., and Ventura, M. (2005). The role of consciousness in memory. *Brains, Minds and Media*, 1(bmm150 (urn:nbn:de:0009-3-1505)).
- [Franklin and Graesser, 2001] Franklin, S. and Graesser, A. (2001). Modelling cognition with software agents. In Moore, J. D. and Stenning, K., editors, *Proceedings of the 23rd Annual Conference of the Cognitive Science Society (CogSci2001)*. Lawrence Erlbaum Associates, Mahwah, N.J.
- [Franklin and McCauley, 2002] Franklin, S. and McCauley, L. (2002). Feelings and emotions as motivators and learning facilitators. In *Architectures for Modelling Emotions, AAAI Spring Symposium, Stanford University, Ca., USA*, AAAI Technical Report SS-04-02. AAAI Press, Menlo Park, Ca., USA.
- [Frith, 1998] Frith, C. (1998). The role of the prefrontal cortex in self-consciousness: the case of auditory hallucinations. In Roberts, A. C., Robbins, T. W., and Weiskrantz, L., editors, *The Prefrontal Cortex: Executive and Cognitive Functions*, pages 195–220. Oxford University Press, Oxford.
- [Fuster, 1995] Fuster, J. M. (1995). *Memory in the cerebral cortex*. MIT Press.
- [Fuster, 2003] Fuster, J. M. (2003). *Cortex and Mind*. Oxford University Press.
- [Gamma et al., 1995] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley, Massachusetts.
- [Gansner and North, 2000] Gansner and North (2000). An open graph visualization system and its applications to software engineering. *SOFTPREX: Software–Practice and Experience*, 30.
- [Garagnani et al., 2002] Garagnani, M., Shastri, L., and Wendelken, C. (2002). A connectionist model of planning as back-chaining search. In D., G. W. and D., S. C., editors, *Proceedings of the 24th Annual Conference of the Cognitive Science Society (CogSci02)*, pages 345–350. Lawrence Erlbaum Associates.

- [Garforth et al., 1999] Garforth, J. P., McHale, S., and Meehan, A. (1999). Adjustably autonomous robotic agents. In Musliner, D. and Pell, P., editors, *Agents with Adjustable Autonomy*, AAAI Spring Symposium 1999, Stanford University, Ca., USA, AAAI Technical Report SS-99-06. AAAI Press, Menlo Park, Ca., USA, ISBN 1-57735-102-9.
- [Garforth et al., 2001a] Garforth, J. P., McHale, S., and Meehan, A. (2001a). Executive attentional control in autonomous robotic systems. In Zhong, N., Liu, J., Ohsuga, S., and Bradshaw, J., editors, *Intelligent Agent Technology Research and Development. Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology*, Maebashi, Japan 23-26 October 2001, pages 479–483. World Scientific Publishing, ISBN 981-02-4706-0.
- [Garforth et al., 2001b] Garforth, J. P., McHale, S., and Meehan, A. (2001b). Problems of attentional behavior in autonomous robotic systems. In Atkins, E. M. and Fesq, L. M., editors, *Robust Autonomy*, AAAI Spring Symposium, Stanford University, Ca., USA, pages 71–75. AAAI Technical Report SS-01-06, AAAI Press, Menlo Park, Ca., USA.
- [Garforth et al., 2003] Garforth, J. P., McHale, S., and Meehan, A. (2003). Neural executive attentional control in robots. In *International Joint Conference on Artificial Intelligence (IJCAI-03)*.
- [Garforth et al., 2004] Garforth, J. P., McHale, S., and Meehan, A. (2004). Executive attention and action selection in a neurally controlled simulated robo. In Smith, L. S., Hussain, A., and Aleksander, I., editors, *Brain Inspired Cognitive Systems (BICS) 2004*.
- [Garforth et al., 2006] Garforth, J. P., McHale, S., and Meehan, A. (2006). Executive attention, task selection and attention-based learning in a neurally controlled simulated robot. *Neurocomputing*, (69):1923–1945.
- [Gathercole, 1994] Gathercole, S. E. (1994). Neuropsychology and working memory: A review. *Neuropsychology*, 8(4):494–505.
- [Gigerenzer et al., 1999] Gigerenzer, G., Todd, P. M., and The ABC Research Group (1999). *Simple Heuristics That Make Us Smart*. Oxford University Press.
- [Gilbert and Shallice, 2001] Gilbert, S. J. and Shallice, T. (2001). Task switching: a PDP model. *Cognitive Psychology*, 44(3):297–337.
- [Glyn Jones and Collins, 1987] Glyn Jones, H. and Collins, B. C. (1987). *A Way of Life: Sheepdog Training, Handling and Trialling*. Farming Press.

- [Gurney et al., 2001a] Gurney, K., Prescott, T. J., and Redgrave, P. (2001a). A computational model of action selection in the basal ganglia i, a new functional anatomy. *Biological Cybernetics*, (84):401–410.
- [Gurney et al., 2001b] Gurney, K., Prescott, T. J., and Redgrave, P. (2001b). A computational model of action selection in the basal ganglia ii, analysis and simulation of behaviour. *Biological Cybernetics*, (84):411–423.
- [Haider and Frensch, 2002] Haider, H. and Frensch, P. A. (2002). Why aggregated learning follows the power law of practice when individual learning does not: Comment on Rickard (1997, 1999), Delaney et al. (1998), and Palmeri (1999). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(2):392–406.
- [Hinton et al., 1993] Hinton, G. E., Plaut, D. C., and Shallice, T. (1993). Simulating brain damage. *Scientific American*, 269(4):76–82.
- [Hinton and Shallice, 1991] Hinton, G. E. and Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review*, 98(1):74–95.
- [Houk et al., 1995] Houk, J. C., Davis, J. L., and Beiser, D. G. (1995). *Models of information processing in the basal ganglia*. MIT press, Cambridge, MA.
- [Houston and Sumida, 1985] Houston, A. and Sumida, B. (1985). A positive feedback model for switching between two activities. *Animal Behaviour*, (33):315–325.
- [James, 1890] James, W. (1890). *The Principles of Psychology, with introduction by George A. Miller*. Harvard University Press, 1983.
- [Kasderidis and Taylor, 2004] Kasderidis, S. and Taylor, J. G. (2004). Attention-based learning. *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, 1:525–530.
- [Kein and Altman, 1992] Kein, J. and Altman, J. S. (1992). *Neuroethology of Action Selection*. MIT Press, Cambridge MA.
- [LaBerge, 1999] LaBerge, D. (1999). Attention. In Rumelhart, D. E. and Martin, B. M., editors, *Cognitive Science: Handbook of Cognition and Perception*. New York: Academic Press.

- [Lachter et al., 2004] Lachter, J., Ruthruff, E., and Forster, K. I. (2004). Forty-five years after broadbent (1958): Still no identification without attention. *Psychological Review*, 111(4):880–913.
- [Lambrinos and Scheier, 1995] Lambrinos, D. and Scheier, C. (1995). Extended braitenburg architectures. AI Lab Technical Report 95-10, University of Zurich.
- [LeDoux and Fellous, 1995] LeDoux, J. E. and Fellous, J.-M. (1995). Emotion and computational neuroscience. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, chapter Part III: Articles, pages 356–359. MIT press.
- [Luria, 1966] Luria, A. R. (1966). *Higher Cortical Functions in Man*. Basic Books, New York.
- [Luria, 1970] Luria, A. R. (1970). The functional organisation of the brain. *Scientific American*, (222):67–78.
- [Maes, 1989] Maes, P. (1989). How to do the right thing. *Connection Science*, (1):291–323.
- [Maes, 1990] Maes, P. (1990). Situated agents can have goals. *Journal for Robotics and Autonomous Systems*, (6).
- [Maes, 1991] Maes, P. (1991). A bottom-up mechanism for behaviour selection in an artificial creature. In Meyer, J. A. and Wilson, S. W., editors, *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behaviour*, pages 238–247. MIT Press/Bradford Books.
- [Maes, 1994] Maes, P. (1994). Modeling adaptive autonomous agents. In *Artificial Life*, volume 1, pages 135–162.
- [Marshall, 2007] Marshall, J. (2007). Future recall: your mind can slip through time. *New Scientist magazine*, (2596):36–40.
- [Mataric, 1996] Mataric, M. J. (1996). Designing and understanding adaptive group behaviour. *Adaptive Behaviour*, 4(1):51–80.
- [McCarthy, 1963] McCarthy, J. (1963). Situations, actions and causal laws. Artificial Intelligence Project Memo 2, Stanford University.

- [McCarthy and Hayes, 1969] McCarthy, J. and Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B. and Michie, D., editors, *Machine Intelligence 4*. Elsevier.
- [McFarland, 1985] McFarland, D. (1985). *Animal Behaviour*. Longman.
- [McFarland and Sibly, 1972] McFarland, D. and Sibly, R. (1972). Unitary drives revisited. *Animal Behaviour*, (20):548–563.
- [McLaughlin, 2000] McLaughlin, B. (2000). *Java and XML*. O'Reilly.
- [Miglino et al., 1995] Miglino, O., Nafasi, K., and Taylor, C. E. (1995). Selection for wandering behavior in a small robot. *Artificial Life*, 2:101–116.
- [Miller, 1956] Miller, G. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, (63):81–97.
- [Miyake and Shah, 1999] Miyake, A. and Shah, P. (1999). Toward unified theories of working memory: Emerging general consensus, unresolved theoretical issues and future research directions. In Miyake, A. and Shah, P., editors, *Models of Working Memory*, chapter 13, pages 442–482. Cambridge University Press.
- [Moray, 1959] Moray, N. (1959). Attention in dichotic listening: Affective cues and the influence of instructions. *Quarterly Journal of Experimental Psychology*, (11):56 – 60.
- [Morén and Balkenius, 2000] Morén, J. and Balkenius, C. (2000). A computational model of emotional learning in the amygdala. In Mayer, J., Berthoz, A., Floreano, D., Roitblat, H., and Wilson, S. W., editors, *From animals to animats 6*, chapter Learning, pages 383–394. MIT press.
- [Morris et al., 2006a] Morris, R., Hitch, G., Graham, K., and Bussey, T. (2006a). Learning and memory. In Morris, R., Tarassenko, L., and Kenward, M., editors, *Cognitive Systems: Information Processing Meets Brain Science*, pages 193–235. Elsevier Academic Press, London.
- [Morris et al., 2006b] Morris, R., Tarassenko, L., and Kenward, M. (2006b). *Cognitive Systems: Information Processing Meets Brain Science*. Elsevier Academic Press, London.

- [Mumford, 1995] Mumford, D. (1995). Thalamus. In Arbib, M. A., editor, *The Handbook of Brain Theory and Neural Networks*, chapter Part III: Articles, pages 981–984. MIT press.
- [Newell, 1990] Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.
- [Newell and Simon, 1976] Newell, A. and Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126.
- [Norman and Shallice, 1986] Norman, D. A. and Shallice, T. (1986). Attention to action: Willed and automatic control of behavior. In Davidson, R. J., Schwartz, G. E., and Shapiro, D., editors, *Consciousness and Self-regulation: Advances in Research and Theory*, volume 4. Plenum Press, New York.
- [Nourbaksh et al., 1995] Nourbaksh, I., Powers, R., and Birchfield, S. (1995). Dervish: An office navigating robot. *AI Magazine*, Summer 1995:53–60.
- [O'Regan and Noë, 2000] O'Regan, J. K. and Noë, A. (2000). Perception, attention and the grand illusion. *PSYCHE*, 6(15).
- [O'Regan and Noë, 2001] O'Regan, J. K. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioural and Brain Sciences*, (24):939–1031.
- [Palmeri, 1999] Palmeri, T. J. (1999). Theories of automaticity and the power law of practice. *journal of experimental psychology. Learning, Memory and Cognition*, (25):543–550.
- [Parkin, 1996] Parkin, A. J. (1996). *Explorations in Cognitive Neuropsychology*. Psychology Press.
- [Piaget, 1936] Piaget, J. (1936). *La naissance de l'intelligence chez l'enfant*. Delachaux et Niestlé, Neuchatel.
- [Plaut and Shallice, 1993] Plaut, D. C. and Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, 10(5):377–500.
- [Posner, 1978] Posner, M. I. . (1978). *Chronometric explorations of mind*. Erlbaum.
- [Prescott et al., 2006] Prescott, T. J., González, F. M. M., Gurney, K., Humphries, M. D., and Redgrave, P. (2006). robot model of the basal ganglia: Behavior and intrinsic processing. *Neural Networks*, 19(1):31–61.

- [Prescott et al., 1999] Prescott, T. J., Redgrave, P., and Gurney, K. (1999). Layered control architectures in robots and vertebrates. *Adaptive Behaviour*, 7(1):99–127.
- [Quinlan, 1991] Quinlan, P. (1991). *Connectionism and psychology: A psychological perspective on new connectionist research*. University of Chicago Press, Chicago.
- [Robbins, 1991] Robbins, T. (1991). Cognitive deficits in schizophrenia and parkinson’s disease: Neural basis and the role of dopamine. In Willner, P. and Scheel-Kruger, J., editors, *The Mesolimbic Dopamine System: From Motivation to Action*, pages 497–526. John Willey.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, (323):533–536.
- [Rumelhart and McClelland, 1986] Rumelhart, D. E. and McClelland, J. L. (1986). PDP models and general issues in cognitive science. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, volume 1. MIT Press/Bradford Books.
- [Shallice, 1988] Shallice, T. (1988). *From Neuropsychology to Mental Structure*. Cambridge University Press.
- [Shallice and Burgess, 1993] Shallice, T. and Burgess, P. (1993). Supervisory control of action and thought selection. In Baddeley, A. and Weiskrantz, L., editors, *Attention: Selection, Awareness, & Control*, chapter 9, pages 171–187. Oxford University Press.
- [Shallice and Burgess, 1998] Shallice, T. and Burgess, P. (1998). *The domain of supervisory processes and the temporal organization of behaviour*, pages 22–35. Oxford University Press, Oxford.
- [Shanahan, 2005] Shanahan, M. (2005). Consciousness, emotion and imagination. In *AISB 2005 Workshop: Next Generation Approaches to Machine Consciousness*, pages 26–35. AISB.
- [Shapiro and Eichenbaum, 1997] Shapiro, M. L. and Eichenbaum, H. B. (1997). Learning and memory: Computational principles and neural mechanisms. In Rugg, M. D., editor, *Studies in cognition-cognitive neuroscience*, pages 77–130. UCL Press.
- [Shastri, 2002] Shastri, L. (2002). Episodic memory and cortico-hippocampal interactions. *Trends in Cognitive Sciences*, 6(4):162–168.
- [Simon, 1957] Simon, H. A. (1957). *Models of Man: Social and Rational*. John Willey.

- [Steels, 1994] Steels, L. (1994). The artificial life roots of artificial intelligence. *Artificial Life*, 1:75–110.
- [Taylor and Fragopanagos, 2003] Taylor, J. and Fragopanagos, N. (2003). Simulations of attention control models in sensory and motor paradigms. *Neural Networks, 2003. Proceedings of the International Joint Conference on*, 1.
- [Taylor and Fragopanagos, 2004] Taylor, J. and Fragopanagos, N. (2004). Modelling the interaction of attention and emotion. *Brain Inspired Cognitive Systems*.
- [Taylor, 2003] Taylor, J. G. (2003). The CODAM model of attention and consciousness. *Neural Networks, 2003. Proceedings of the International Joint Conference on*, 1.
- [Tononi and Cirelli, 2003] Tononi, G. and Cirelli, C. (2003). Sleep and synaptic homeostasis: a hypothesis. *Brain Research Bulletin*, 62(2):143–150.
- [Tononi and Cirelli, 2006] Tononi, G. and Cirelli, C. (2006). Sleep function and synaptic homeostasis. *Sleep Medicine Reviews*, 10(1):49–62.
- [Tulving, 1993] Tulving, E. (1993). What is episodic memory? *Current Directions in Psychological Science*, 2(3):67–70.
- [Tyrrell, 1992] Tyrrell, T. (1992). Defining the action selection problem. In *Proc. of the fourteenth annual conf. of the Cognitive Society*. Lawrence Erlbaum Associates.
- [Tyrrell, 1993a] Tyrrell, T. (1993a). *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh.
- [Tyrrell, 1993b] Tyrrell, T. (1993b). Journal of adaptive systems. *The use of hierarchies for action selection*, 1(4):387–419.
- [van Zomeren and Spikman, 2003] van Zomeren, E. and Spikman, J. (2003). Assessment of attention. In Halligan, P. W., Kischka, U., and Marshall, J. C., editors, *Handbook of Clinical Neuropsychology*, chapter 5, pages 73–87. Oxford University Press.
- [Walter, 1953] Walter, W. G. (1953). *The Living Brain*. Gerald Duckworth & Co. Ltd.

- [Zell et al., 1993] Zell, A., Mamier, G., Hübner, R., Schmalzl, N., Sommer, T., and Vogt, M. (1993). SNNS: An efficient simulator for neural nets. In Schwetman, H. D., Walrand, J. C., Bagchi, K. K., and DeGroot, D., editors, *MASCOTS '93, Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, January 17-20, 1993, La Jolla, San Diego, CA, USA*, pages 343–346. The Society for Computer Simulation.
- [Zurada, 1992] Zurada, J. M. (1992). *Artificial Neural Systems*. West Publishing Company.

Appendix A

11. An Overview of the Sheep & Sheepdogs

11.1. Sheep & Sheepdogs

Prior to the commencement of this research, experiments were designed to observe the behaviours of a fully autonomous robot flock of sheep facing an autonomous robot sheepdog commanded by a human handler [Glyn Jones and Collins, 1987]. Though real sheepdogs know only four basic commands, working with a handler, alone or in pairs (a brace) it was discovered that the dog's repertoire of behaviours as well as the behaviour of the sheep was very complex. Several real machines were created, programmed and observed in relation to this work, and experiences with these machines, (their behavioural issues, amongst others, leading to the creation of Attentional Architecture) ensured that the simulation environments and the simulated robots were grounded.

11.2. Autonomous Robots

11.2.1. Sheep 1, Sheep 2 & Sheepdog

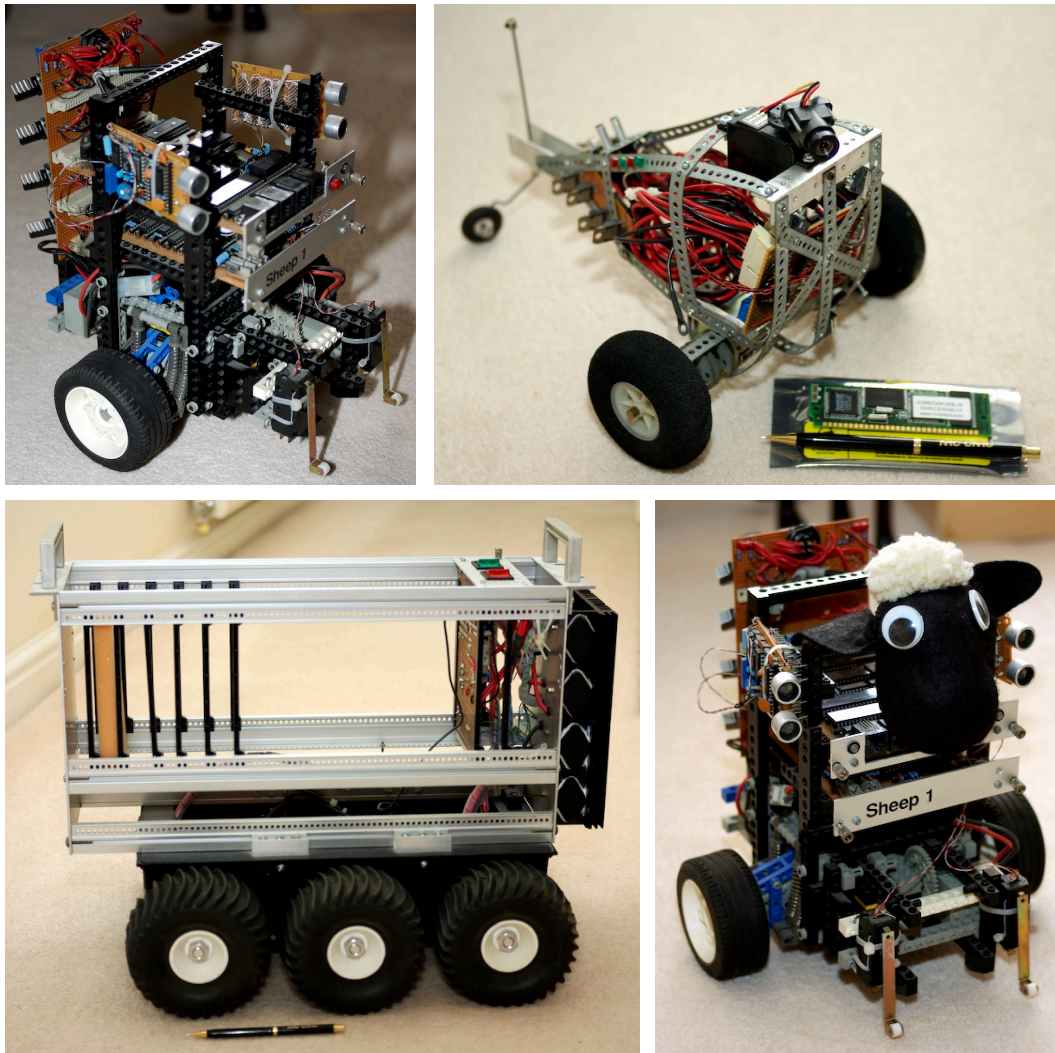


figure 11.1 Sheep 1 (top left), Sheep 2 (top right) and Sheepdog (bottom left). The chassis used for the Sheepdog is waterproof for outdoor use and chasing wildlife (e.g. ducks, lambs etc.)

Appendix B

12. XML for the Attentional Architecture

12.1. Attentional Architecture, Controller XML

The following is the XML used to build the instance of the Attentional Architecture lesioned in Chapter 7.

```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- Created by jpg on February 19, 2007, 11:40 AM -->

<!DOCTYPE network SYSTEM "nuroxml_network.dtd">

<!-- This Neuro XML network file builds the demonstration implementation of the Attentional Architecture
as described in the July 2007 Thesis - Chapter 7

(C) Jason Garforth/OU 2006/7

-->

<network>

  <!-- This LINE commands the Network Builder to build a network composed of
neurons connected to each sensor and actuator that have been registered
by the machine - NOTE that the network is automatically built
-->
  <nuronet name="robo_hardware" type="SIMROBOT" dupl="1" thread="false"/>

  <!-- Build the simulated Chemical Interface -->
  <nuronet name="chemical" type="CHEMICAL" dupl="1" thread="true"/>

  <!-- Episodic Memory -->
  <nurocollection name="episodic_memory" type="collection" thread="true">

    <nurocollection name="layer1" type="collection" thread="false">

      <nurocollection name="chemical" type="collection" thread="false">
        <nurocollection name="food_present" type="collection" thread="false">
          <nuronet name="control" cluster="control_bottom_up_food.net" type="SNNS"
            dupl="1" thread="false"/>
          <nuronet name="recognition" cluster="recognition_chem_percep.net"
            type="SNNS" dupl="1" thread="false"/>
        </nurocollection>

        <nurocollection name="home_present" type="collection" thread="false">
          <nuronet name="control" network="control_bottom_up.net" type="SNNS"
            dupl="1" thread="false"/>
          <nuronet name="recognition" network="recognition_chem_percep.net"
            type="SNNS" dupl="1" thread="false"/>
        </nurocollection>
      </nurocollection>

      <nurocollection name="gripper" type="collection" thread="false">
        <nurocollection name="sense" type="collection" thread="false">
          <nuronet name="control" network="control_bottom_up.net" type="SNNS"
            dupl="1" thread="false"/>
          <nuronet name="recognition" network="recognition_gripper_percep.net"
            type="SNNS" dupl="1" thread="false"/>
        </nurocollection>
      </nurocollection>

    </nurocollection>

    <nurocollection name="layer2" type="collection" thread="false">

      <nurocollection name="touching_food" type="collection" thread="false">
        <nuronet name="control" network="control_bottom_up.net" type="SNNS" dupl="1"
          thread="false"/>
        <nuronet name="recognition" network="recognition_touching_food.net" type="SNNS"
          dupl="1" thread="false"/>
      </nurocollection>

      <nurocollection name="got_food" type="collection" thread="false">
        <nuronet name="control" network="control_bottom_up.net" type="SNNS" dupl="1"
          thread="false"/>
        <nuronet name="recognition" network="recognition_simple_and.net" type="SNNS"
          dupl="1" thread="false"/>
      </nurocollection>

    </nurocollection>

  </nurocollection>

</network>
```

```

<!-- Working Memory -->
<nurocollection name="working_memory" type="collection" thread="true">

    <!-- Some (All?) recogniser nodes are implemented in two parts:
        A node control network which interacts with the SAS, Associative &
        Contention Layers
        A recogniser network which maps input patterns to output patterns
    -->
    <nurocollection name="layer1" type="collection" thread="false">
        <nurocollection name="collect_food" type="collection" thread="false">

            <nuronet name="control" network="control_observer.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" network="recognition_sequence4.net" type="SNNS"
                dupl="1" thread="false"/>

        </nurocollection>
    </nurocollection>

</nurocollection>

<!-- Behaviours Implemented by this machine -->
<nurocollection name="procedural_memory" type="collection" thread="true">

    <!-- Low Level Nodes:
        These nodes implement the sensor to actuator mappings and may be though of as
        being the "motor cortex", it is these behaviours which are responsible for
        movement
        and some sequencing
    -->
    <nurocollection name="layer1" type="collection" thread="false">

        <!-- Orient TO and MOVE towards FOOD / HOME -->
        <nurocollection name="orient_chemical" type="collection" thread="false">
            <nuronet name="control" network="control_top_down.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" network="recognition_orient_chem.net" type="SNNS"
                dupl="1" thread="false"/>
        </nurocollection>

        <!-- Pick UP FOOD -->
        <nurocollection name="pick_up_food" type="collection" thread="false">
            <nuronet name="control" network="control_top_down.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" network="recognition_gripper_pickup.net" type="SNNS"
                dupl="1" thread="false"/>
            <nuronet name="feedback" network="recognition_and_feedback.net" type="SNNS"
                dupl="1" thread="false"/>
        </nurocollection>

        <!-- Drop OFF FOOD -->
        <nurocollection name="drop_off_food" type="collection" thread="false">
            <nuronet name="control" network="control_top_down.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" network="recognition_gripper_drop.net" type="SNNS"
                dupl="1" thread="false"/>
            <nuronet name="feedback" network="recognition_and_feedback.net" type="SNNS"
                dupl="1" thread="false"/>
        </nurocollection>
    </nurocollection>

    <nurocollection name="layer2" type="collection" thread="false">

        <nurocollection name="find_food" type="collection" thread="false">
            <nuronet name="control" network="control_top_down.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" network="recognition_sequence4.net" type="SNNS"
                dupl="1" thread="false"/>
        </nurocollection>

        <nurocollection name="return_home" type="collection" thread="false">
            <nuronet name="control" network="control_top_down.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="recognition" network="recognition_sequence4.net" type="SNNS"
                dupl="1" thread="false"/>
        </nurocollection>
    </nurocollection>

</nurocollection>

<!-- Contention Scheduler, Note how this should be layered -->
<nurocollection name="contention_scheduler" type="collection" thread="true">

    <nurocollection name="episodic_memory" type="collection" thread="false">
        <nurocollection name="layer1" type="collection" thread="false">

```

```
<nuronet name="orient_path" network="contention_scheduler_4.net" type="SNNS"
  dupl="1" thread="true"/>
</nurocollection>
</nurocollection>

<nurocollection name="working_memory" type="collection" thread="false">
  <nurocollection name="layer1" type="collection" thread="false">
    <nuronet name="collect_food" network="contention_scheduler_4.net" type="SNNS"
      dupl="1" thread="false"/>
  </nurocollection>
</nurocollection>

<nurocollection name="procedural_memory" type="collection" thread="false">
  <nurocollection name="layer1" type="collection" thread="false">
    <nuronet name="motor" network="contention_scheduler_4.net" type="SNNS" dupl="1"
      thread="true"/>
    <nuronet name="gripper" network="contention_scheduler_4.net" type="SNNS"
      dupl="1" thread="false"/>
  </nurocollection>
</nurocollection>

</nurocollection>

<!-- Limbic System - used as an interface to the chemical subsystem -->
<nurocollection name="limbic_system" type="collection" thread="true">

  <nurocollection name="hippocampus" type="collection" thread="false">

    <nuronet name="food_present" network="thalamic_gateway.net" type="SNNS" dupl="1"
      thread="true"/>

  </nurocollection>

</nurocollection>

<!-- Supervisor Attentional System:
The SAS is the principal object of this simulation and is currently composed of three
distinct areas:

SAS_monitor:

SAS_generate:

SAS_modulate:

-->
<nurocollection name="sas" type="collection" thread="true">

  <nurocollection name="monitor_modulate" type="collection" thread="false">

    <!-- Supervisorory Attentional Monitor System -->

    <nurocollection name="episodic_memory" type="collection" thread="false">

      <nurocollection name="layer1" type="collection" thread="false">

        <nuronet name="food_present" network="sas_monitor_modulator.net"
          type="SNNS" dupl="1" thread="false"/>
        <nuronet name="home_present" network="sas_monitor_modulator.net"
          type="SNNS" dupl="1" thread="false"/>

      </nurocollection>
    </nurocollection>

    <nurocollection name="working_memory" type="collection" thread="false">

      <nurocollection name="layer1" type="collection" thread="false">

        <nuronet name="collect_food" network="sas_monitor_modulator.net"
          type="SNNS" dupl="1" thread="false"/>

      </nurocollection>
    </nurocollection>

    <nurocollection name="procedural_memory" type="collection" thread="false">

      <nurocollection name="layer1" type="collection" thread="false">

        <nuronet name="orient_chemical" network="sas_monitor_modulator.net"
          type="SNNS" dupl="1" thread="false"/>
        <nuronet name="pick_up_food" network="sas_monitor_modulator.net"
          type="SNNS" dupl="1" thread="false"/>
        <nuronet name="drop_off_food" network="sas_monitor_modulator.net"
          type="SNNS" dupl="1" thread="false"/>

      </nurocollection>
    </nurocollection>

  </nurocollection>
</nurocollection>
```

```

</nurocollection>

<nurocollection name="layer2" type="collection" thread="false">

    <nuronet name="find_food" network="sas_monitor_modulator.net" type="SNNS"
        dupl="1" thread="false"/>
    <nuronet name="return_home" network="sas_monitor_modulator.net"
        type="SNNS" dupl="1" thread="false"/>

</nurocollection>

</nurocollection>

</nurocollection>
</nurocollection>

<!-- Thalamus Output System:
    The thalamus regulates all output to and from the cortex, in our model it is used by the
    contention scheduler to protect shared information pathways. The thalamus like the
    contention scheduler is highly stratified by functional use. Some Models (Prescott et al)
    use the thalamus as a sensory "attentional" filter - this is also used in the attentional
    architecture as part of the reactive system
-->
<nurocollection name="thalamus_gateway" type="collection" thread="true">
    <nurocollection name="episodic_memory" type="collection" thread="false">

        <nurocollection name="orient_path_food" type="collection" thread="false">
            <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="gate_3" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="gate_4" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
        </nurocollection>

        <nurocollection name="orient_path_home" type="collection" thread="false">
            <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="gate_3" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
            <nuronet name="gate_4" network="thalamic_gateway.net" type="SNNS" dupl="1"
                thread="false"/>
        </nurocollection>

    </nurocollection>

    <nurocollection name="procedural_memory" type="collection" thread="false">
        <nurocollection name="motor" type="collection" thread="false">

            <nurocollection name="orient_chemical" type="collection" thread="false">
                <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_3" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_4" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
            </nurocollection>

            <nurocollection name="pick_up_food" type="collection" thread="false">
                <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_3" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_4" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
            </nurocollection>

            <nurocollection name="drop_off_food" type="collection" thread="false">
                <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_3" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_4" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
            </nurocollection>

        </nurocollection>
    </nurocollection>

```

```

        <nurocollection name="gripper" type="collection" thread="false">
            <nurocollection name="pick_up_food" type="collection" thread="false">
                <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
            </nurocollection>
            <nurocollection name="drop_off_food" type="collection" thread="false">
                <nuronet name="gate_1" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
                <nuronet name="gate_2" network="thalamic_gateway.net" type="SNNS" dupl="1"
                    thread="false"/>
            </nurocollection>
        </nurocollection>
    </nurocollection>
</nurocollection>

    <!-- These Mappings make use of SheepWorld WildCards - they will match entire names -->
    <nurocollection name="episodic_procedural_home_food_mappings" type="nuomapping" thread="false">

        <!-- Map the Robots Chemical Sensors into the lowest layer of episodic memory (Perception
Layer) -->
        <!-- FOOD -->
        <nuomap source="robo_hardware.chemfood_1_reading_s"
            target="episodic_memory.layer1.chemical.food_present.recognition.inp1"/>
        <nuomap source="robo_hardware.chemfood_2_reading_s"
            target="episodic_memory.layer1.chemical.food_present.recognition.inp2"/>
        <nuomap source="robo_hardware.chemfood_3_reading_s"
            target="episodic_memory.layer1.chemical.food_present.recognition.inp3"/>
        <nuomap source="robo_hardware.chemfood_4_reading_s"
            target="episodic_memory.layer1.chemical.food_present.recognition.inp4"/>
        <!-- HOME -->
        <nuomap source="robo_hardware.chemhome_1_reading_s"
            target="episodic_memory.layer1.chemical.home_present.recognition.inp1"/>
        <nuomap source="robo_hardware.chemhome_2_reading_s"
            target="episodic_memory.layer1.chemical.home_present.recognition.inp2"/>
        <nuomap source="robo_hardware.chemhome_3_reading_s"
            target="episodic_memory.layer1.chemical.home_present.recognition.inp3"/>
        <nuomap source="robo_hardware.chemhome_4_reading_s"
            target="episodic_memory.layer1.chemical.home_present.recognition.inp4"/>

        <!-- Map the recogniser network, salience output (avail) food/home into the salience input of
the control network (Simulates a Habitual Response) -->
        <!-- Food is an addiction -->
        <nuomap source="episodic_memory.layer1.chemical.food_present.recognition.avail"
            target="episodic_memory.layer1.chemical.food_present.control.pattern_match" weight="3"/>
        <nuomap source="episodic_memory.layer1.chemical.home_present.recognition.avail"
            target="episodic_memory.layer1.chemical.home_present.control.pattern_match" weight="1"/>

        <!-- Map the episodic memory cluster output into procedural memory (the reactive machine) via
the thalamus chemical_path -->
        <nuomap source="episodic_memory.layer1.chemical.food_present.recognition.out*"
            target="thalamus_gateway.episodic_memory.orient_path_food.gate_#1.inp2"/>
        <nuomap source="episodic_memory.layer1.chemical.home_present.recognition.out*"
            target="thalamus_gateway.episodic_memory.orient_path_home.gate_#1.inp2"/>

        <!-- Map the thalamus output into procedural memory and the orient_chemical cluster -->
        <nuomap source="thalamus_gateway.episodic_memory.orient_path_food.gate_*.out1"
            target="procedural_memory.layer1.orient_chemical.recognition.inp#1" weight="1.5"/>
        <nuomap source="thalamus_gateway.episodic_memory.orient_path_home.gate_*.out1"
            target="procedural_memory.layer1.orient_chemical.recognition.inp#1" weight="1.5"/>

        <!-- Map the contention scheduler output for the chemical pathway to the disinhibition inputs
in the thalamus -->
        <nuomap source="contention_scheduler.episodic_memory.layer1.orient_path.out1"
            target="thalamus_gateway.episodic_memory.orient_path_food.gate_*.inp1"/>
        <nuomap source="contention_scheduler.episodic_memory.layer1.orient_path.out2"
            target="thalamus_gateway.episodic_memory.orient_path_home.gate_*.inp1"/>

        <!-- Map the recogniser node expression_level output from episodic memory to the contention
scheduler -->
        <nuomap source="episodic_memory.layer1.chemical.food_present.control.expression_level"
            target="contention_scheduler.episodic_memory.layer1.orient_path.inp1" weight="3"/>
        <nuomap source="episodic_memory.layer1.chemical.home_present.control.expression_level"
            target="contention_scheduler.episodic_memory.layer1.orient_path.inp2" weight="0.75"/>

        <!-- Thalamo-cortical feedback from the orient_path scheduler to episodic memory food/home -->
        <nuomap source="contention_scheduler.episodic_memory.layer1.orient_path.out1"
            target="episodic_memory.layer1.chemical.food_present.control.scheduler_feedback"/>
        <nuomap source="contention_scheduler.episodic_memory.layer1.orient_path.out2"
            target="episodic_memory.layer1.chemical.home_present.control.scheduler_feedback"/>

        <!-- Map the recogniser node in procedural memory into the contention schedulers for the
motors -->

```

```

    <nuomap source="procedural_memory.layer1.orient_chemical.control.expression_level"
target="contention_scheduler.procedural_memory.layer1.motor.inp1"/>
    <!-- And its thalamo-cortical feedback back into procedural memory -->
    <nuomap source="contention_scheduler.procedural_memory.layer1.motor.out1"
target="procedural_memory.layer1.orient_chemical.control.scheduler_feedback"/>

    <!-- Map the Contention scheduler motor outputs (disinhibitors) for orient_chemical into the
    Thalamus -->
    <nuomap source="contention_scheduler.procedural_memory.layer1.motor.out1"
target="thalamus_gateway.procedural_memory.motor.orient_chemical.gate *.inp1"/>
    <!-- Map the output of procedural memory recogniser node orient_chemical into the thalamus -->
    <nuomap source="procedural_memory.layer1.orient_chemical.recognition.out*"
target="thalamus_gateway.procedural_memory.motor.orient_chemical.gate_#1.inp2"/>

    <!-- Map the Motor Outputs from the Thalamus to the robots actuators -->
    <nuomap source="thalamus_gateway.procedural_memory.motor.*.gate_1.out1"
target="robo_hardware.motor1_left_a"/>
    <nuomap source="thalamus_gateway.procedural_memory.motor.*.gate_2.out1"
target="robo_hardware.motor1_leftdir_a"/>
    <nuomap source="thalamus_gateway.procedural_memory.motor.*.gate_3.out1"
target="robo_hardware.motor1_right_a"/>
    <nuomap source="thalamus_gateway.procedural_memory.motor.*.gate_4.out1"
target="robo_hardware.motor1_rightdir_a"/>

</nurocollection>

<!-- Working Memory connections into procedural and episodic memory -->
<!-- It is suggested that episodic connections (from working memory) go through procedural memory -->
<nurocollection name="workingmemory" type="nuomapping" thread="false">

    <!-- collect_food -->
    <nuomap source="working_memory.layer1.collect_food.recognition.out1"
target="procedural_memory.layer1.orient_chemical.control.excitation" weight="0.5"/>
    <nuomap source="working_memory.layer1.collect_food.recognition.out1"
target="episodic_memory.layer1.chemical.food_present.control.excitation" weight="0.5"/>

    <!-- pick_up_food -->
    <!-- Connect working memory (preparatory set) to procedural memory to activate this node -->
    <nuomap source="working_memory.layer1.collect_food.recognition.out2"
target="procedural_memory.layer1.pick_up_food.control.excitation"/>

    <!-- go home -->
    <nuomap source="working_memory.layer1.collect_food.recognition.out3"
target="procedural_memory.layer1.orient_chemical.control.excitation" weight="0.5"/>
    <nuomap source="working_memory.layer1.collect_food.recognition.out3"
target="episodic_memory.layer1.chemical.home_present.control.excitation" weight="0.5"/>

    <!-- drop food off -->
    <!-- Connect working memory (preparatory set) to procedural memory to activate this node -->
    <nuomap source="working_memory.layer1.collect_food.recognition.out4"
target="procedural_memory.layer1.drop_off_food.wrap.excitation"/>

    <!-- Layer 2 of procedural memory -->
    <nuomap source="working_memory.layer1.collect_food.recognition.out1"
target="procedural_memory.layer2.find_food.control.excitation" weight="0.5"/>
    <nuomap source="working_memory.layer1.collect_food.recognition.out3"
target="procedural_memory.layer2.return_home.control.excitation" weight="0.5"/>

</nurocollection>

<!-- Episodic Memory layer two (compound episodes) - touching_food -->
<nurocollection name="episodic_memory_touching_food" type="nuomapping" thread="false">

    <!-- Map the GRIPPER sensors into episodic memory... -->
    <nuomap source="robo_hardware.gripper1_Touching_S"
target="episodic_memory.layer1.gripper.sense.recognition.inp1"/>
    <nuomap source="robo_hardware.gripper1_Grabbed_S"
target="episodic_memory.layer1.gripper.sense.recognition.inp2"/>

    <nuomap source="episodic_memory.layer1.gripper.sense.recognition.out1"
target="episodic_memory.layer1.gripper.sense.control.pattern_match" weight="5.0"/>

    <!-- This is a bottom up recogniser node, with no links to a contention scheduler, loop back
    the salience input to the scheduler_feedback -->
    <nuomap source="episodic_memory.layer1.gripper.sense.control.pattern_match"
target="episodic_memory.layer1.gripper.sense.control.scheduler_feedback"/>

    <!-- Map the TOUCHING FOOD episode into working memroy for PICK UP FOOD -->
    <nuomap source="episodic_memory.layer1.gripper.sense.control.salience"
target="episodic_memory.layer2.touching_food.recognition.inp1"/>
    <nuomap source="episodic_memory.layer1.chemical.food_present.salience"
target="episodic_memory.layer2.touching_food.recognition.inp2"/>

    <nuomap source="episodic_memory.layer2.touching_food.recognition.out1"
target="episodic_memory.layer2.touching_food.control.pattern_match" weight="5.0"/>

```

```

    <!-- This is a bottom up recogniser node, with no links to a contention scheduler, loop back
    the salience input to the scheduler_feedback -->
    <nuomap source="episodic_memory.layer2.touching_food.control.pattern_match"
    target="episodic_memory.layer2.touching_food.control.scheduler_feedback"/>

    <!-- This should cause the transition from ORIENT_FOOD to PICK_UP_FOOD -->
    <nuomap source="episodic_memory.layer2.touching_food.control.salience"
    target="working_memory.layer1.collect_food.recognition.inp2"/>

</nurocollection>

<!-- Procedural memory mappings for pick up food -->
<nurocollection name="proceduralmemorypickupfood" type="nuomapping" thread="false">

    <!-- To pick up food, it has to be there and the gripper has to be touching it... -->
    <nuomap source="episodic_memory.layer2.touching_food.control.salience"
    target="procedural_memory.layer1.pick_up_food.recognition.inp1"/>
    <nuomap source="episodic_memory.layer1.gripper.sense.control.salience"
    target="procedural_memory.layer1.pick_up_food.recognition.inp2"/>

    <!-- Pick up food has to hold the machine steady and own the grippers - hence it has to use
    two contention schedulers... -->
    <nuomap source="procedural_memory.layer1.pick_up_food.control.expression_level"
    target="contention_scheduler.procedural_memory.layer1.motor.inp2"/>
    <nuomap source="procedural_memory.layer1.pick_up_food.control.expression_level"
    target="contention_scheduler.procedural_memory.layer1.gripper.inp1"/>
    <!-- And the thalamo-cortical feedback... at present we use a specialist summation network
    where two or more schedulers are used -->
    <!-- In this case we have to collate the feedback from two areas of the contention scheduler
    as both motors and grippers are requested -->
    <nuomap source="contention_scheduler.procedural_memory.layer1.motor.out2"
    target="procedural_memory.layer1.pick_up_food.feedback.inp1"/>
    <nuomap source="contention_scheduler.procedural_memory.layer1.gripper.out1"
    target="procedural_memory.layer1.pick_up_food.feedback.inp2"/>
    <nuomap source="procedural_memory.layer1.pick_up_food.feedback.out1"
    target="procedural_memory.layer1.pick_up_food.control.scheduler_feedback"/>

    <!-- The out2 OUTPUT is ZERO to HOLD the machine still whilst were grabbing the food -->
    <nuomap source="procedural_memory.layer1.pick_up_food.recognition.out2"
    target="thalamus_gateway.procedural_memory.motor.pick_up_food.gate_*.inp2"/>

    <!-- Map the contention scheduler output into the thalamus for the motors -->
    <nuomap source="contention_scheduler.procedural_memory.layer1.motor.out2"
    target="thalamus_gateway.procedural_memory.motor.pick_up_food.gate_*.inp1"/>

    <!-- Map the contention scheduler output into the thalamus for the gripper -->
    <nuomap source="contention_scheduler.procedural_memory.layer1.gripper.out1"
    target="thalamus_gateway.procedural_memory.gripper.pick_up_food.gate_*.inp1"/>

    <!-- Map the procedural memory output into the thalamus for the grippers -->
    <nuomap source="procedural_memory.layer1.pick_up_food.recognition.out*"
    target="thalamus_gateway.procedural_memory.gripper.gate_#.inp2"/>

    <!-- Map the Gripper Outputs from the Thalamus to the robots actuators -->
    <nuomap source="thalamus_gateway.procedural_memory.gripper.gate_1.out1"
    target="robo_hardware.gripper1_Grab_A"/>
    <nuomap source="thalamus_gateway.procedural_memory.gripper.gate_2.out1"
    target="robo_hardware.gripper1_Drop_A"/>

    <!-- We now use priming of episodic memory, by working/procedural memory to ensure that we
    have got the food -->
    <nuomap source="episodic_memory.layer1.gripper.sense.control.salience"
    target="episodic_memory.layer2.got_food.recognition.inp1"/>
    <nuomap source="working_memory.layer1.collect_food.recognition.out2"
    target="episodic_memory.layer2.got_food.recognition.inp2"/>

    <nuomap source="episodic_memory.layer2.got_food.recognition.out1"
    target="episodic_memory.layer2.got_food.control.pattern_match"/>

    <!-- This is a bottom up recogniser node, with no links to a contention scheduler, loop back
    the salience input to the scheduler_feedback -->
    <nuomap source="episodic_memory.layer2.got_food.control.pattern_match"
    target="episodic_memory.layer2.got_food.control.scheduler_feedback"/>

    <!-- And use this to trigger the next state - go home -->
    <nuomap source="episodic_memory.layer2.got_food.control.salience"
    target="working_memory.layer1.collect_food.recognition.inp3"/>

    <!-- Add in the links from layer 2 to sequence layer 1 for the top down recogniser nodes
    'find_food' and 'return_home' -->
    <nuomap source="procedural_memory.layer2.find_food.recognition.out1"
    target="procedural_memory.layer1.orient_chemical.control.excititation" weight="0"/>
    <nuomap source="procedural_memory.layer2.find_food.recognition.out1"
    target="procedural_memory.layer1.pick_up_food.control.excititation" weight="0"/>

    <!-- Loop Back the expression_level to scheduler_feedback - a contention scheduler isn't used
    for these nodes yes -->

```

```

    <nuomap source="procedural_memory.layer2.find_food.control.expression_level"
target="procedural_memory.layer2.find_food.control.scheduler_feedback"/>

    <nuomap source="procedural_memory.layer2.return_home.recognition.out1"
target="procedural_memory.layer1.orient_chemical.control.excitation" weight="0"/>
    <nuomap source="procedural_memory.layer2.return_home.recognition.out1"
target="procedural_memory.layer1.drop_off_food.control.excitation" weight="0"/>

    <!-- Loop Back the expression_level to scheduler_feedback - a contention scheduler isn't used
for these nodes yes -->
    <nuomap source="procedural_memory.layer2.return_home.control.expression_level"
target="procedural_memory.layer2.return_home.control.scheduler_feedback"/>

    <!-- Priming from Procedural Memory to Episodic Memory - food and home -->
    <nuomap source="procedural_memory.layer2.find_food.recognition.saliency"
target="episodic_memory.layer1.chemical.food_present.excitation" weight="0"/>
    <nuomap source="procedural_memory.layer2.find_food.recognition.saliency"
target="episodic_memory.layer1.chemical.pick_up_food.excitation" weight="0"/>

</nurocollection>

<!-- We have a SAS node for each episodic/working/procedural memory recogniser node that uses a
contention scheduler -->
<nurocollection name="sasmappings" type="nuromapping" thread="false">

    <!-- Episodic Memory SAS Mappings, at present only food_present & home_present use the
Contention Scheduler -->
    <nurocollection name="food_present" type="nuromapping" thread="false">
        <nuomap source="episodic_memory.layer1.chemical.food_present.control.excitation"
target="sas.monitor_modulate.episodic_memory.layer1.food_present.excitation" weight="1.0"/>
        <nuomap source="episodic_memory.layer1.chemical.food_present.control.expression_level"
target="sas.monitor_modulate.episodic_memory.layer1.food_present.expression_level" weight="2.0" />
        <nuomap
source="episodic_memory.layer1.chemical.food_present.control.scheduler_feedback"
target="sas.monitor_modulate.episodic_memory.layer1.food_present.scheduler_feedback" weight="1.0"/>
        <nuomap
source="sas.monitor_modulate.episodic_memory.layer1.food_present.modulation_cs"
target="contention_scheduler.episodic_memory.layer1.orient_path.sinp1" weight="1.0"/>
    </nurocollection>

    <!-- Episodic Memory SAS Mappings, at present only food_present & home_present use the
Contention Scheduler -->
    <nurocollection name="home_present" type="nuromapping" thread="false">

        <nuomap source="episodic_memory.layer1.chemical.home_present.control.excitation"
target="sas.monitor_modulate.episodic_memory.layer1.home_present.excitation"/>
        <nuomap source="episodic_memory.layer1.chemical.home_present.control.expression_level"
target="sas.monitor_modulate.episodic_memory.layer1.home_present.expression_level"/>

        <nuomap
source="episodic_memory.layer1.chemical.home_present.control.scheduler_feedback"
target="sas.monitor_modulate.episodic_memory.layer1.home_present.scheduler_feedback"/>

        <nuomap source="sas.monitor_modulate.episodic_memory.home_present.modulation_out"
target="contention_scheduler.episodic_memory.layer1.orient_path.sinp2"/>
    </nurocollection>
-->

    <!-- nurocollection name="sasworkingmemorymaps" type="nuromapping" thread="false" -->
    <!-- There are none at this time... collect_food doesn't use a contention scheduler... yet...
-->
    <!-- /nurocollection -->

    <!-- SAS Procedural Memory mappings -->
    <!--
    <nurocollection name="sasproceduralmemorymaps" type="nuromapping" thread="false">
-->

        <!-- Procedural Memory - orient_chemical
        <nuomap source="procedural_memory.layer1.orient_chemical.control.excitation"

target="sas.monitor_modulate.procedural_memory.layer1.orient_chemical.excitation"/>
        <nuomap source="procedural_memory.layer1.orient_chemical.control.expression_level"

target="sas.monitor_modulate.procedural_memory.layer1.orient_chemical.expression_level"/>
        <nuomap source="procedural_memory.layer1.orient_chemical.control.scheduler_feedback"

target="sas.monitor_modulate.procedural_memory.layer1.orient_chemical.scheduler_feedback"/>
        <nuomap source="sas.monitor_modulate.procedural_memory.orient_chemical.moulation_out"
target="contention_scheduler.procedural_memory.layer1.motor.sinp1"/>
        -->

        <!-- Procedural Memory - pick up food, note it uses two contention schedulers
(motor/gripper) hence the two outputs from its SAS Monitor/Modulator

```

```
<nuromap source="procedural_memory.layer1.pick_up_food.control.excitation"
target="sas.monitor_modulate.procedural_memory.layer1.pick_up_food.excitation"/>

<nuromap source="procedural_memory.layer1.pick_up_food.control.expression_level"
target="sas.monitor_modulate.procedural_memory.layer1.pick_up_food.expression_level"/>

<nuromap source="procedural_memory.layer1.pick_up_food.control.scheduler_feedback"
target="sas.monitor_modulate.procedural_memory.layer1.pick_up_food.scheduler_feedback"/>

<nuromap source="sas.monitor_modulate.procedural_memory.pick_up_food.moulation_out"
target="contention_scheduler.procedural_memory.layer1.motor.sinp2"/>
<nuromap source="sas.monitor_modulate.procedural_memory.pick_up_food.moulation_out"
target="contention_scheduler.procedural_memory.layer1.gripper.sinp1"/>
-->
<!--
</nurocollection>
-->

</nurocollection>

<!-- Housekeeping, enable all the contention schedulers when requested by the robot to do so... -->
<nurocollection name="enabler" type="nuromapping" thread="false">
  <nuromap source="robo_hardware.controll1_Contention_Enable_S"
target="contention_scheduler.*.*.*.enable" weight="1.0" tag="Contention Scheduler
Control Enable"/>

  <nuromap source="robo_hardware.controll1_reset_all_s"
target="working_memory.layer1.collect_food.recognition.inp5" weight="1.0" tag="Working
Memory Reset"/>
</nurocollection>

<!-- Mappings to/from the hippocampus to support emotion based learning -->
<nurocollection name="hippocampusconnections" type="nuromapping" thread="false">

  <!-- Mappings for the Hippocampus based learning subsystem -->
  <nuromap source="sas.monitor_modulate.episodic_memory.layer1.food_present.modulation_out"
target="limbic_system.hippocampus.food_present.inp1" weight="2.0" tag="SAS Learning
Inp"/>
  <nuromap source="episodic_memory.layer1.chemical.food_present.control.expression_level"
target="limbic_system.hippocampus.food_present.inp2" weight="2.0" tag="Episodic Food
Present Inp" />
  <nuromap source="limbic_system.hippocampus.food_present.out1" target="chemical.chemical_1"
weight="25" tag="Hipp chemical Out"/>

</nurocollection>

<!-- The following is an inhibitory link between touching_food and orient_to_food -->
<sensitize modulator="1" threshold="0.70" time="80">
  <!-- <nuromap source="episodic_memory.layer1.chemical.home_present.control.salience"
target="episodic_memory.layer1.chemical.food_present.control.excitation" weight="0"
type="2"/> -->
  <nuromap source="episodic_memory.layer2.touching_food.control.salience"
target="episodic_memory.layer1.chemical.food_present.control.excitation" weight="0"
type="2"/>
</sensitize>

</network>
```

12.2. Attentional Architecture, Controller Document Type Definition (DTD)

The following is the Document Type Definition file used by SheepWorld3 to build the SAX 2, XML parser in Java for the XML defined in *section 12.1*.

```
<?xml version='1.0' encoding='UTF-8'?>

<!--
  TODO define vocabulary indentification
  PUBLIC ID: -//vendor//vocabulary//EN
  SYSTEM ID: http://server/path/simple21\_test.dtd

--><!--
  An example how to use this DTD from your XML document:

  <?xml version="1.0"?>

  <!DOCTYPE network SYSTEM "attentional_architecture.dtd">
  <network>
  ...
  </network>
-->

<!ELEMENT sensitize (nuromap|sensitize)*>
<!ATTLIST sensitize
  tag CDATA #IMPLIED
  time CDATA #IMPLIED
  refractory CDATA #IMPLIED
  threshold CDATA #IMPLIED
  chemical CDATA #IMPLIED
  modulator CDATA #IMPLIED
  >

<!ELEMENT nuromap EMPTY>
<!ATTLIST nuromap
  type CDATA #IMPLIED
  target CDATA #IMPLIED
  source CDATA #IMPLIED
  weight CDATA #IMPLIED
  tag CDATA #IMPLIED
  >

<!ELEMENT nuronet (nuronet)*>
<!ATTLIST nuronet
  network CDATA #IMPLIED
  cluster CDATA #IMPLIED
  thread CDATA #IMPLIED
  dupl CDATA #IMPLIED
  type CDATA #IMPLIED
  name CDATA #IMPLIED
  >
```



```
<!ELEMENT nuromap (nuronet|nuromap|nuocollection)*>
<!ATTLIST nuromap
    thread CDATA #IMPLIED
    type CDATA #IMPLIED
    name CDATA #IMPLIED
>

<!ELEMENT network (sensitize|nuromap|nuronet|nuocollection)*>
<!ELEMENT controller (sensitize|nuromap|nuronet|nuocollection)*>
```

12.3. Creating SheepWorld3, the Attentional Architecture and this Thesis

This thesis and the simulations that it presents were created using several software platforms and packages, a special thank you to the creators of:

- Apple (Mac OS X, Apple Pages, Mail, Preview)
- Adobe (Acrobat)
- Sun Microsystems (Java, Netbeans, templates and layout taken from open source troff)
- Microsoft (Word 2004, Excel 2004)
- IDraw, still working on BSD UNIX, 20 years after it was written
- Inkscape, Graphvis, Latexit, Omnigraffle, Gimp, Bibtex, PDFLab
- GNU/Linux for not losing a single file in 10 years

Made on a Mac

